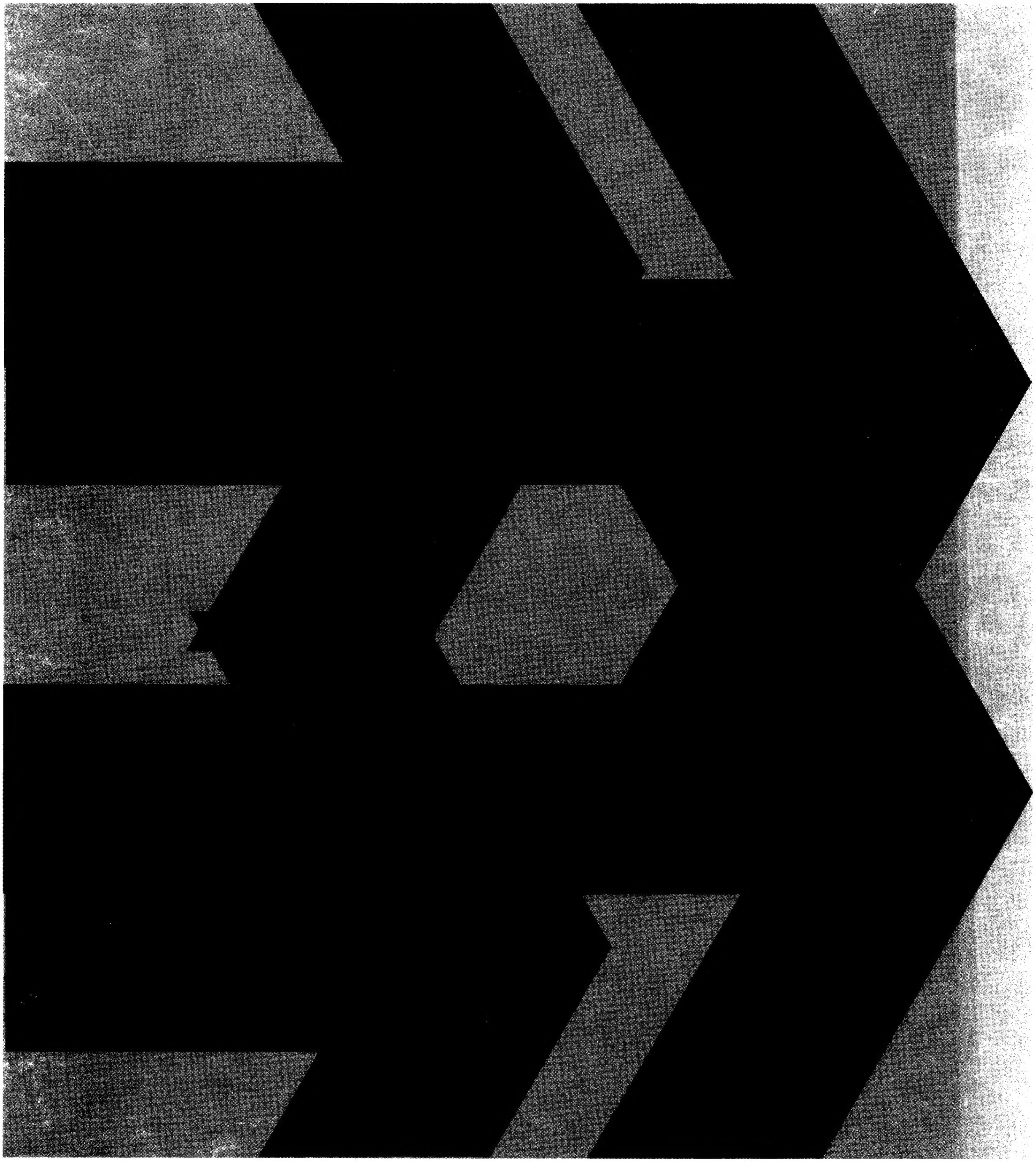
HEWLETT PACKARD COMPUTER CURRICULUM

COLLEGE AND UNIVERSITY SERIES

CLASSICAL STATISTICAL MECHANICS

by Herbert D. Peckham



HEWLETT PACKARD COMPUTER CURRICULUM

COLLEGE AND UNIVERSITY SERIES

CLASSICAL STATISTICAL MECHANICS

by **HERBERT D. PECKHAM**
Gavilan College
Gilroy, California

edited by **CHRISTINE DOERR**
Hewlett-Packard

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD
CUPERTINO, CALIFORNIA 95014

HP 5951-5669

Printed: April 1974

The Hewlett-Packard Computer Curriculum Series represents the published results of a Curriculum Development project sponsored by the Data Systems Division of Hewlett-Packard Company. This project is under the directorship of Jean H. Danver.

This material is designed to be used with any Hewlett-Packard system with the BASIC programming language such as the 9830A, Educational BASIC, and the 2000 and 3000 series systems.

The Hewlett-Packard Company makes no warranty, expressed or implied, and assumes no responsibility in connection with the operation of the program material attached hereto.

PREFACE

This unit was written to give the student of statistical mechanics an opportunity to bring the power of the computer to bear on this fascinating subject. The computer can be used as a very effective tool to probe the concepts involved.

It is assumed that this unit is being used as a supplementary text in a course in statistical mechanics. The only requirements besides being enrolled in or having taken such a course is some familarity with BASIC language programming and access to a computer.

Many of the exercises involve writing or modifying computer programs. This is intentional and follows from the strategy that one learns more and becomes more fundamentally involved by writing programs than by merely interacting with programs already prepared. The required level of skill in programming is not high. Students not familar with BASIC should be able to do fine after a couple of hours of instruction or study on their own.

Students and instructors as well should avoid falling into the trap of thinking that here is a right or best method to solve a specific problem. Certainly some programs may be more elegant or efficient than others. However, to become obsessed with intricate solutions or flashy details of programming is to draw attention away from the physics. Remember that the purpose of this unit is not to teach about computers, but to use computers to learn about physics.

This unit is not meant to be a text, and that no effort has been made to provide a complete treatment of statistical mechanics. Excellent texts are available and should be used for completeness. This unit is designed to reinforce such a text and follows closely the approach used by Reif[3] and also draws heavily upon the unique description of statistical mechanics by Sherwin[4].

ABOUT THE AUTHOR

Herb Peckham's contributions towards furthering the use of computers in physics instruction are too substantial to give a thorought treatment here.

His publications include the book, *Computers, BASIC and Physics,* published by Addison Wesley, and, by the Hewlett-Packard Computer Curriculum Project, four physics units for high school and college use, and a publication on *Air Pollution.* He is also the author of *Computer Graphics: Three Dimensional Projections,* due to be released by Hewlett-Packard in the summer of 1974.

In addition to his publishing activities, Herb has spoken about computers in physics instruction on a number of occasions including engagements at the Conferences on Computers in the Undergraduate Curricula, AIP/AAPT joint meetings, ACM meetings, and numerous local gatherings. He is a member of the physics committee for CONDUIT, a NSF project devoted to distributing computer curriculum.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

MACROSTATES AND MICROSTATES

We will begin this unit with a brief definition of *macrostates* and *microstates* since this is at the heart of statistical mechanics. The prefixes macro and micro are defined with respect to atomic dimensions. A microscopic object has dimensions of the order of magnitude of the atomic diameter (about 10^{-8} cm). On the other hand, a macroscopic object has dimensions very much larger than the atomic size. Macroscopic objects are "people-sized" and constitute the everyday world of our experience.

By *macrostate,* we mean a complete description of a macroscopic system using macroscopic parameters. For example, we might describe a container of gas in terms of its volume, pressure, and temperature. These are large scale measurements which are readily available to us. A *microstate* is, as we would expect, a description of a microscopic system using microscopic parameters. For example, we could specify the mass, position, and velocity of each of the particles in the container of gas at some instant. It is obvious that this very large quantity of microscopic information is generally not available to us. However, it is also clear that in some fashion the macrostate is functionally dependent upon the microstate.

Our fundamental problem in statistical mechanics is to find a way to understand the behavior of macroscopic systems containing very large numbers of particles (of the order of magnitude of Avogadro's Number, or about 10^{24}) by reasoning from known first principles at the atomic level. It is easy to see that this is hopeless if approached from the point of view of classical mechanics. Given one particle moving under the influence of known forces, it is a routine matter to determine its motion and momentum as functions of time. With a computer, we can do the same thing for several particles. However, with 10^{24} particles it is clear that the task cannot be accomplished with the techniques of classical mechanics, even with the largest computer. Another approach is required.

The necessary bridge between the microstate and corresponding macrostate is provided by statistical mechanics. This is a very well understood and complete branch of physics that has been developed in the last 100 years. With statistical mechanics we can, in fact, predict descriptions of macrosystems from microscopic information. It is thus interesting to note that solutions are available for the two extremes — one particle, or very many particles — but that both classical mechanics and statistical mechanics fail in the region between.

HISTORICAL PERSPECTIVE

Quite often, the terms thermodynamics and statistical thermodynamics are carelessly substituted for statistical mechanics, causing confusion in the minds of many students. From a historical point of view, the differences between these terms are interesting and provide valuable insight.

Thermodynamics is generally understood to be defined by the following four laws:

1. Zeroth Law

   Two systems, each of which is in equilibrium with a third system, are in equilibrium with each other.

2. First Law

   $\Delta\bar{E} = W + Q$,

   where $\Delta\bar{E}$ is the change in mean energy of a system, W is the work done on the system, and Q is the heat transferred to the system.

3. Second Law

   $dS = \frac{dQ}{T}$, and $\Delta S >= 0$,

   where S is the entropy of a system, and T is the absolute temperature at which an amount of heat dQ is transferred to the system.

4. Third Law

   As $T \rightarrow 0, S \rightarrow S_0$,

   where S_0 is a constant independent of the system.

These laws which define thermodynamics are completely macroscopic in nature. They were discovered and applied long before a comprehensive theory of atoms and molecules was available. Note that the laws make no reference to the detailed structure of the system, nor is there any element of probability present. Functionally, however, thermodynamics is a useful tool even though it cannot bridge the gap between the microstate and macrostate.

If we add the notion of probability to thermodynamics, we define statistical thermodynamics. Probability is introduced through the statistical relation

$$P \propto e^{S/k} ;$$

where P is the probability of observing a macrostate with entropy S, and k is the Boltzmann constant.

Finally, if we connect the entropy S to the number of microstates accessible to the system, we have statistical mechanics. Statistical mechanics is therefore the broadest subject, with statistical thermodynamics and thermodynamics being progressively more narrowly defined. The key to statistical mechanics is the relation

$$S = k \ln(W)$$

which is credited to the German physicist Ludwig Boltzmann (1844-1906). The significance of this relation is underlined by the fact that it is carved into the Boltzmann memorial in the Central Cemetery in Vienna. A more modern version is

$$S = k \ln(\Omega)$$

where k is the Boltzmann constant and Ω is the number of microstates accessible to the system. Josiah Williard Gibbs (1839-1903), the first noteworthy American theoretical physicist, used this modern version and the concept of statistical ensembles to construct a very general and beautiful framework for statistical mechanics that remains valid in the most modern applications.

THE ROLE OF THE COMPUTER

It is important to establish the role of the computer with respect to this unit on statistical mechanics. Most of the essential ideas of statistical mechanics can be demonstrated by examining the behavior of systems containing from 20 to 100 elements. The computer is required to carry out the lengthy calculations which usually would preclude this type of investigation. Consequently, the computer is a tool which gives us the leverage we need to get at the ideas of statistical mechanics. Using the computer as a device to simulate the random processes involved, we can uncover those ideas which have fundamental importance. It is precisely those ideas which are most often slighted in favor of a cookbook-formula approach to statistical mechanics.

CHAPTER TWO: PROBABILITY, STATISTICAL MEASUREMENT, AND RANDOM PROCESSES

The concepts of probability and random processes are fundamental to any modern treatment of statistical mechanics. Systems reacting to the influence of blind chance behave in predictable ways. Consequently, we must carefully consider the topics of probability and statistical measurements if we hope to understand the behavior of statistical systems.

LAWS OF PROBABILITY

We will define the laws of probability in terms of states. By state we mean a complete description of a system or event. The fundamental law of probability defined in terms of states is

$$P(A) = \frac{\text{Number of states which imply A}}{\text{Total number of possible states}} \quad . \tag{1}$$

P(A) is read as "The probability of A." If all the states imply A, then P(A) = 1. If none of the states imply A, then P(A) = 0. Thus, all probabilities fall in the range 0 to 1 inclusive.

Two examples will illustrate the fundamental law:

1. If we deal a card from a well-shuffled bridge deck of cards, what is the probability of getting a King?

   $$P(K) = \frac{\text{Number of states corresponding to a King}}{\text{Total number of possible states}} = \frac{4}{52} \quad .$$

2. If a pair of dice is thrown, what is the probability that the faces uppermost will add to 7? In this case we can describe a state with a number pair. The first number is the uppermost face on die 1, the second number refers to die 2. Thus, (3,1) indicates a 3 on die 1, and a 1 on die 2. Note that (3,1) and (1,3) are different states. It is easy to enumerate the states which correspond to a sum of seven. They are (1,6), (6,1), (2,5), (5,2), (3,4), and (4,3) for a total of 6 states. Since to each of the six faces of die 1 we could have six faces of die 2, there are a total of 36 outcome states possible. Thus

   $$P(\text{Sum of } 7) = \frac{6}{36} \quad .$$

We can use (1) to define other laws of probability. Suppose we want to know the probability of *either* event A or event B. The law which gives us this probability is

$$P(A \text{ or } B) = P(A) + P(B) \quad . \tag{2}$$

Returning to the deck of cards, we can easily compute the probability of dealing either a heart or a club:

$$\begin{aligned} P(\text{Heart or Club}) &= P(\text{Heart}) + P(\text{Club}) \\ &= 13/52 + 13/52 = 26/52. \end{aligned}$$

To follow the idea expressed by (2) to an important conclusion, suppose that A, B, C, and D are the *only* possible outcome states for a system. It follows that

$$P(\text{A or B or C or D}) = P(A) + P(B) + P(C) + P(D) = 1 \quad .$$

This is equivalent to the statement that *something* must happen. We generalize this concept to

$$P(\text{One of the possible outcome states occurring}) = 1 \tag{3}$$

Another type of situation we must be able to handle is that of an event A followed by an event B. If the two events are independent (the occurrence of the second event is not related to the occurrence of the first) we have

$$P(\text{A and B}) = P(A)P(B) \quad . \tag{4}$$

However, if the two events are dependent (the occurrence of the second event is related to the occurrence of the first) the desired relation is

$$P(\text{A and B}) = P(A)P(B|A) \quad . \tag{5}$$

P(B|A) is read as "The probability of B given the occurrence of A."

Some examples should clarify the difference between dependent and independent events.

1. If a deck of cards is shuffled, a card is dealt, and then returned to the deck which is again shuffled. A second card is dealt. What is the probability of getting a spade followed by a King? Clearly the two events are independent since the occurrence of the first event cannot affect the second. Therefore we use (4).

$$\begin{aligned} P(\text{Spade and King}) &= P(\text{Spade})P(\text{King}) \\ &= (13/52)\,(4/52) = 1/52 \end{aligned}$$

2. Now suppose that two cards are dealt in succession. Now the events are dependent since the occurrence of the first event bears on the second, and we should use (5). What is the probability of getting the Ace of hearts, then any other heart?

$$\begin{aligned} P(A_{\text{hearts}} \text{ and Heart}) &= P(A_{\text{hearts}})P(\text{Heart}|A_{\text{hearts}}) \\ &= (1/52)(12/51) = 12/2652 \end{aligned}$$

A final idea concerning probability which is needed involves combinations. For example, suppose we want to know the probability of getting two heads and three tails if five coins are flipped at once. Any of the following states correspond to our desired event:

State Number	State
1	H H T T T
2	H T H T T
3	H T T H T
4	H T T T H
5	T H H T T
6	T H T H T
7	T H T T H
8	T T H H T
9	T T H T H
10	T T T H H

Therefore, we want

$$P(\text{State 1 or State 2 or} \cdot\cdot\cdot \text{ or State 10}).$$

However, it can be seen that

$$P(\text{State 1}) = P(\text{State 2}) = \cdot\cdot\cdot\cdot = P(\text{State 10}).$$

Consequently, the probability of two heads and three tails when five coins are tossed is given by

$$P(2H \text{ and } 3T) = 10\, P(H\&H\&T\&T\&T) = 10\,(1/2)^5.$$

In this case it was easy to enumerate all the possible states. In general, however, this is not true and we must be able to compute the number of states. If N_r is the number of states corresponding to N items taken n at a time, then

$$N_r = \frac{N!}{n!\,(N-n)!} \quad . \tag{6}$$

If N = 5, and n = 2

$$N_r = \frac{5!}{2!3!} = \frac{(4)(5)}{(1)(2)} = 10 \ ,$$

which agrees with our previous result.

MEASURE OF CENTRAL TENDENCY AND DISPERSION

We need an economical way to characterize a set of measurements made on a system. Of course, one way would be to simply list all the numbers. However, this is inefficient and does not make it possible to compare different sets of measurements directly. On the other hand, all data can be described in terms of two parameters: the *central tendency* of the data, and the *dispersion* of the data about the point of central tendency.

The *measure of central tendency* is defined as a point about which the data tends to group. This point is simply the arithmetic average of the data which is called the mean. There are two other measures of central tendency (the median and mode) but we will not need these here. You can pursue the subject in any introductory text on statistics if you are interested.

There are two ways to compute a mean. They are defined in quite different ways but both are very useful. If we are given a set of data x_1, x_2, x_3, • • •, x_i, • • •, x_n the arithmetic mean is defined as

$$\bar{x} = \frac{1}{n}\Sigma\, x_i \ . \tag{7}$$

If you are not familiar with the sigma (Σ) notation in (7) it is merely a concise way to indicate that all the values of x_i are to be summed. The mean is read as "x bar."

As indicated above, there is a second definition of the mean that is very useful in statistical mechanics. Suppose that measurements of a system will yield a set of numbers x_1, x_2, x_3, • • •, x_i, • • •, x_n. Moreover, suppose we know that the system yields the measurements x_i with probability $P(x_i)$ for i = 1,2,3, • • •, n. Then, the mean value of the measurements of the system is

$$\bar{x} = \Sigma\, x_i P(x_i) \ . \tag{8}$$

A simple example can illustrate the point that the mean does not completely describe a set of data. Consider the two sets of data

$$\{49,50,51\} \text{ and } \{0,25,50,75,100\} \ .$$

Both sets of data have a mean of 50. However, it is obvious that the sets are quite different. The second set is spread out about the mean much more than the first. What we need is a measure of this spreading out or dispersion.

We might try to measure the dispersion by subtracting the mean from each element of data and then summing these differences which are called *deviations.* For the two data sets above we compute

$$(x_i - \bar{x})$$

which gives

$$\{-1,0,1\} \text{ and } \{-50,-25,0,25,50\} \ .$$

If the deviations in both sets are summed we see that

$$\text{Sum of Deviations} = \Sigma\,(x_i - \bar{x}) = 0 \ . \tag{9}$$

Thus, the deviation cannot be used to measure dispersion. As a matter of fact, (9) is an alternative way to define the mean.

A simple change is all that is needed to produce our desired measure of dispersion. Instead of summing the deviations, we sum the square of the deviations and then divide by the number of measurements. The result is called the *variance.* The variance is therefore the mean of the square of the deviations of a set of data. This is usually called the *mean square deviation.*

$$\text{Variance} = s^2 = \frac{1}{n}\Sigma\,(x_i - \bar{x})^2 \tag{10}$$

For reasons that we will not explore here, it turns out that (10) is a biased estimate of the variance. The unbiased estimate is

$$\text{Variance} = s^2 = \frac{1}{(n-1)}\,\Sigma\,(x_i - \bar{x})^2 \ .$$

Since you may encounter this slightly different form, you should know the reason for the difference. However, we will use the first form given by (10).

The variance can be computed using (10). However, $\bar{x}$ is required which is a separate calculation. Some straightforward algebra is all that is needed to convert (10) to a form that involves only the raw data.

$$s^2 = \frac{n \Sigma x_i^2 - (\Sigma x_i)^2}{n^2} \tag{11}$$

Since we do not have to compute $\bar{x}$, (11) is generally easier to use.

Just as we have a relationship for $\bar{x}$ in terms of P(x), we should have the same type of relationship for s^2. This is

$$s^2 = \Sigma P(x_i)\ (x_i - \bar{x})^2 \ , \tag{12}$$

or the equivalent form which requires only raw data

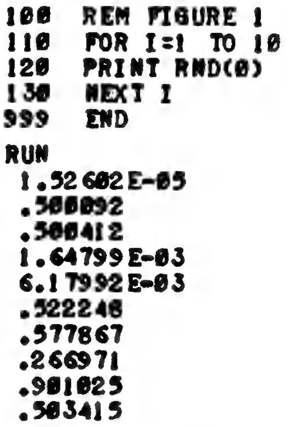
$$s^2 = \Sigma x_i^2\ P(x_i) - (\Sigma x_i P(x_i))^2 \ . \tag{13}$$

You may have wondered why in equations (10) through (13) the variance is written as s^2. The reason is that the square root of the variance is the *standard deviation* which is usually designated by s. Recall that the variance is called the mean square deviation. Consequently, the standard deviation is called the *root mean square deviation.*

MONTE CARLO METHOD

The Monte Carlo Method is characterized by the utilization of random numbers. Many different types of phenomena can be investigated using the method. We are particularly interested in this method since the random processes which are fundamental in statistical mechanics can be investigated using Monte Carlo techniques.

The heart of the Monte Carlo Method is a sequence of numbers which appears to be drawn at random from a given distribution. Most computers have a random number generator which can be used to produce such a sequence. A BASIC program and printout to generate 10 random numbers is shown in Figure 1.

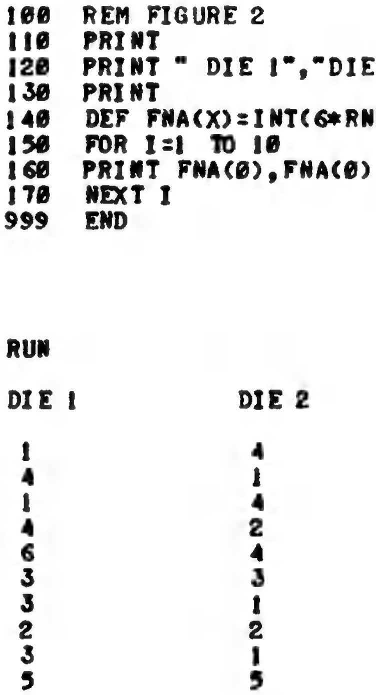


```
100  REM FIGURE 1
110  FOR I=1  TO 10
120  PRINT RND(0)
130  NEXT I
999  END
RUN
 1.52602E-05
 .500092
 .500412
 1.64799E-03
 6.17992E-03
 .522248
 .577867
 .266971
 .901025
 .503415
```

Figure 1 — Random Number Generation

Two interesting characteristics are seen in the printout. First, there seems to be no particular pattern in the numbers. Second, the numbers all fall between 0 and 1. The random number generator in BASIC is designed to conform to these two features. The numbers occur randomly in the range 0 to 1. All numbers in the range have an equal probability of occurring. In the program itself, the random numbers are generated in line 120. The argument of the RND function is taken to be zero. It is a dummy argument and does not affect the random numbers that are generated.

Suppose we want to use the random number generator to simulate throwing a pair of dice. To do this we need a sequence of integers randomly chosen in the range 1 to 6. A program to simulate 10 tosses of a pair of dice is shown in Figure 2.



```
1ØØ  REM FIGURE 2
11Ø  PRINT
12Ø  PRINT " DIE 1","DIE 2"
13Ø  PRINT
14Ø  DEF FNA(X)=INT(6*RND(X)+1)
15Ø  FOR I=1 TO 1Ø
16Ø  PRINT FNA(Ø),FNA(Ø)
17Ø  NEXT I
999  END


RUN

DIE 1          DIE 2

 1              4
 4              1
 1              4
 4              2
 6              4
 3              3
 3              1
 2              2
 3              1
 5              5
```

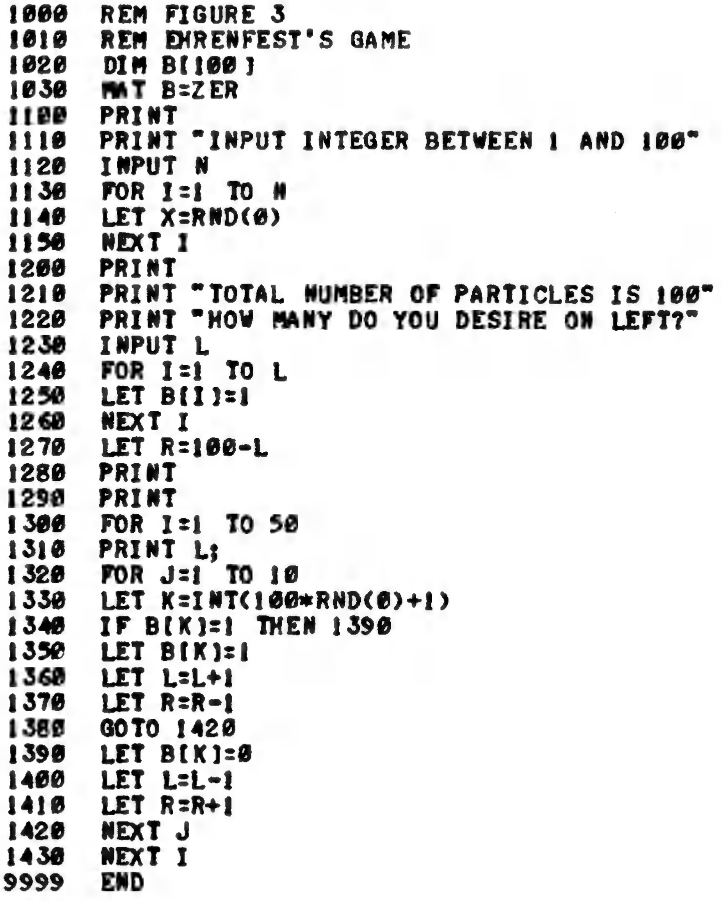
Figure 2 — Dice Simulation

The desired set of random integers is generated in line 140. With a little thought you can see the purpose of each part of the line, and can generate any other type of sequence desired.

THE EHRENFEST GAME

At a somewhat primitive level, some of the ideas of random processes and their effect upon statistical systems can be illustrated with a game credited to the German physicist Paul Ehrenfest (1880-1933). In the simplest form, N particles are divided between two halves of a container. A move is made by selecting one of the N particles at random, and moving it to the other side of the container. A large number of moves are made in this fashion. If, initially, all the particles are in the same side of the container, the overwhelming probability is that the flow of particles will be toward the empty side. However, as the number of particles increases on the empty side, so does the probability that some particles will start moving in the reverse direction. This gives rise to the statistical scatter which is an important characteristic of macrosystems.

A simple BASIC program to play the Ehrenfest game is given in Figure 3. We will use this program in some of the exercises, so it is important that you understand its features. The program is organized to handle 100 particles. Each particle is represented by an element of the array B. The array is dimensioned in line 1020, and has all its elements set equal to zero in line 1030. The program segment from lines 1110 to 1150 starts the needed sequence of random numbers at a different point each time the program is run. This is done by "throwing away" an arbitrary quantity of random numbers. The program segment from lines 1210 through 1270 enables you to set the desired number of particles on the left side of the container.



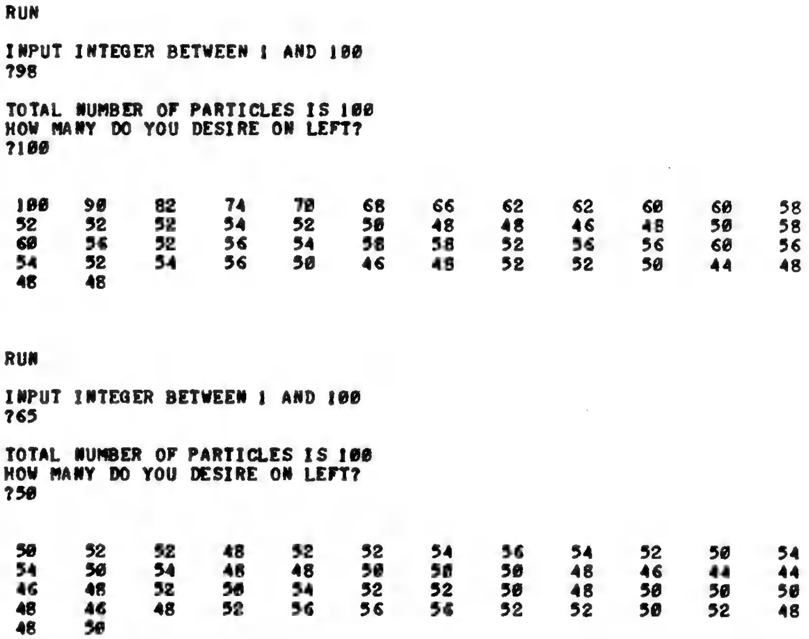
```
1000  REM FIGURE 3
1010  REM EHRENFEST'S GAME
1020  DIM B[100]
1030  MAT B=ZER
1100  PRINT
1110  PRINT "INPUT INTEGER BETWEEN 1 AND 100"
1120  INPUT N
1130  FOR I=1 TO N
1140  LET X=RND(0)
1150  NEXT I
1200  PRINT
1210  PRINT "TOTAL NUMBER OF PARTICLES IS 100"
1220  PRINT "HOW MANY DO YOU DESIRE ON LEFT?"
1230  INPUT L
1240  FOR I=1 TO L
1250  LET B[I]=1
1260  NEXT I
1270  LET R=100-L
1280  PRINT
1290  PRINT
1300  FOR I=1 TO 50
1310  PRINT L;
1320  FOR J=1 TO 10
1330  LET K=INT(100*RND(0)+1)
1340  IF B[K]=1 THEN 1390
1350  LET B[K]=1
1360  LET L=L+1
1370  LET R=R-1
1380  GOTO 1420
1390  LET B[K]=0
1400  LET L=L-1
1410  LET R=R+1
1420  NEXT J
1430  NEXT I
9999  END
```

Figure 3 — Program For The Ehrenfest Game

Of course this should be equal to or less than 100. If a particle is on the left, its value in the array B is 1. If on the right, it has the value 0. Thus, $B_{78} = 0$ indicates that particle number 78 is on the right side of the container.

The core of the game is contained in lines 1300 through 1430. The program is set to print out the number of particles on the left side 50 times, with 10 moves to be made between each printout. The random nature of the process is introduced in line 1330. K is randomly assigned an integer value in the range 1 to 100. K is then used as a subscript to locate a particle (an element in the array B). The particle is tested to see which side it is on, then moved to the other side, and the counters indicating the number of particles on each side are adjusted accordingly.

Figure 4 shows two typical printouts for the game. The first started with all the particles on the left side. Note the steady progress towards the equilibrium value of 50. The second case started with 50 particles on each side and illustrates the scatter about equilibrium.



```
RUN

INPUT INTEGER BETWEEN 1 AND 100
?98

TOTAL NUMBER OF PARTICLES IS 100
HOW MANY DO YOU DESIRE ON LEFT?
?100

 100    90     82     74     70     68     66     62     62     60     60     58
 52     52     52     54     52     50     48     48     46     48     50     58
 60     56     52     56     54     58     58     52     56     56     60     56
 54     52     54     56     50     46     48     52     52     50     44     48
 48     48

RUN

INPUT INTEGER BETWEEN 1 AND 100
?65

TOTAL NUMBER OF PARTICLES IS 100
HOW MANY DO YOU DESIRE ON LEFT?
?50

 50     52     52     48     52     52     54     56     54     52     50     54
 54     50     54     48     48     50     50     50     48     46     44     44
 46     48     52     50     54     52     52     50     48     50     50     50
 48     46     48     52     56     56     56     52     52     50     52     48
 48     50
```

Figure 4 — Printouts For The Ehrenfest Game

EXERCISE 1 — Flipping Coins

Suppose that 5 fair coins are simultaneously flipped. Calculate the probability that exactly n of the coins will be heads for all possible values of n. Now write a BASIC program to simulate tossing the 5 coins. Arrange for your program to simulate 500 tosses of the coins. Each 50 tosses print out the frequency of occurrence of n heads divided by the total number of tosses. Compare the sequence of computer results to the theoretical values you calculated. What kind of generalizations can you make?

EXERCISE 2 — Dishonest Coins

Repeat Exercise 1, but assume the coins are dishonest with P(H) = 2/3 and P(T) = 1/3.

EXERCISE 3 – A Data Set

Write a program to compute the mean and variance of a set of data. Assume that the data is contained in DATA statements, and that the last piece of data is followed by the "flag value" 9999. Use the flag to detect when all the data has been processed. Test your program on the following data.

9.1, 10.3, 9.8, 9.9, 10.1,

EXERCISE 4 – A Frequency Distribution

Modify the program of Exercise 3 to compute the mean and variance of the data given in a frequency distribution. Test your program on the following distribution.

x	*Frequency*
15	1
14	3
13	10
12	7
11	5
10	2

EXERCISE 5 – A Probability Distribution

Suppose we know that a variable x occurs with the following probability:

x	*P(x)*
1	.04
2	.08
3	.12
4	.16
5	.20
6	.16
7	.12
8	.08
9	.04

If we measure a great number of values of x, what will the mean and variance of our measurements be? Write a BASIC program to compute and print out the mean and variance.

EXERCISE 6 – A Random Walk

Write a BASIC program to simulate a 100-step random walk. Assume that the walk begins at x = 0, and that the probability of a step forward is the same as a step backward.

EXERCISE 7 – A Biased Random Walk

A very interesting problem has been suggested by Weinstock[5]. Consider a sequence of squares numbered 0 through 10. A drunk is placed on square N. Assume that the probability of the drunk moving to a higher-numbered *square at each step is 0.6, and the probability of his moving to a lower-numbered square is 0.4. What should N be such that it is equally likely that the drunk will step off either end of the sequence of squares? Write a program to simulate many walks with different starting points to investigate the problem. The results are surprising!*

EXERCISE 8 – The Ehrenfest Game

The game presented in Figure 3 assumes that the particles are moved between two sides with equal volumes. The result is that the equilibrium value of particles in each side is one-half the total. Rewrite the program to account for unequal volumes. Test your program to ensure that the proper equilibrium number of particles in each side is obtained.

EXERCISE 9 – Relative Fluctuations

By combining programs already written, we can investigate the fluctuation of the number of particles about the equilibrium point in the Ehrenfest game. In particular, we are interested in $s/\bar{x}$, or the ratio of the standard deviation to the mean. Write a program which will play the Ehrenfest game and output this quantity. Arrange your program so that you can compute $S/\bar{x}$ for various total numbers of particles. What should happen to this quantity as the number of particles increases? Check your answer on the computer.

CHAPTER THREE: SIMULATED CRYSTAL STRUCTURE

A COMPUTER GAME

A simple game has been described by Black[1] which is very useful in the study of statistical mechanics. Let us first examine the game in a very simple form. Then the connection to microphysics will be established.

Suppose we have an ordinary checkerboard with 8 squares on a side. We distribute N pennies on the board in any way desired. If $n_{i,j}$ is the number of pennies on the square with row number i, and column number j, then it follows that

$$\Sigma\Sigma\, n_{i,j} = N \ . \tag{14}$$

The rules of the game are as follows:

a) Select a square at random and call it A. If square A has at least 1 penny on it, proceed to b). Otherwise, select another square at random.

b) Select a second square at random and call this square B. If B turns out to be the same square as A, select another square at random.

c) Move a penny from square A to B. Repeat the entire process beginning with a).

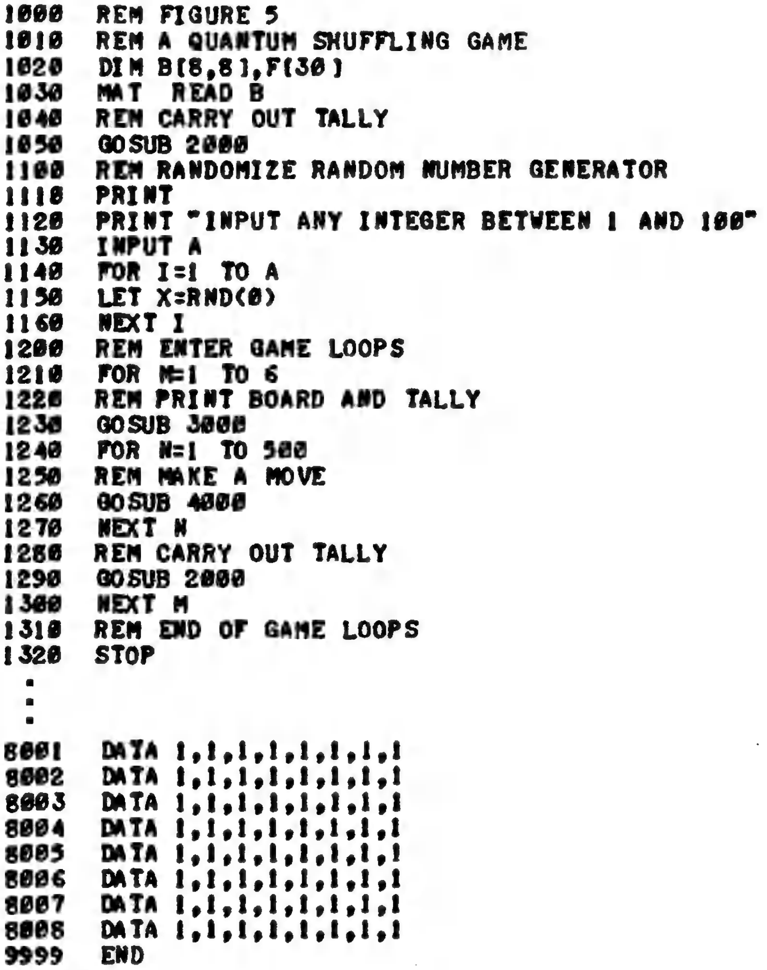
EXERCISE 10 – A Checkerboard Game

Write a BASIC program to generate a table of integers randomly chosen in the range 1 to 8 inclusive. Begin at some point in the table and select two integers to locate square A, and the next two integers to locate square B. As the game is played, continue to use the number table to randomly locate squares on the checkerboard. Distribute 64 pennies on the checkerboard in any way you desire. Play the game for at least 100 moves. See if you can make any generalizations about what is happening. Try different starting distributions to see if this has any effect upon your generalizations.

By now you have certainly concluded that playing the game by hand will limit the amount of information we can gain. We should also point out that there is nothing particularly important about the 8 by 8 array upon which we are playing the game. It could be played upon any size array.

As advertised, we will now make the connection between the game and statistical mechanics. The game is a simulation of energy transfer in a crystal structure. In this simulation, we have a two-dimensional crystal, but could equally well develop a three-dimensional model if desired. Each square represents an atom at a point in the crystal. We assume that the atom can be described by a harmonic oscillator that is quantized, that is, the oscillator can absorb or give off energy only in fixed amounts called *quanta.* The movement of pennies on the checkerboard thus corresponds to transfer of quanta between harmonic oscillators in the simulated crystal. The number of pennies on a square corresponds to the total energy of the oscillator at that point in the crystal. No pennies corresponds to an energy of $h\nu_o/2$, the lowest energy possible for a quantized oscillator.

A computer program to play the shuffling game is given in Figure 5. The logic of the overall program is shown in Figure 5(a). The subroutines which are needed are contained in Figures 5(b) through 5(d). Since we will be using this program extensively, we will discuss it in some detail to permit you to understand the purpose of each segment and the relation to the game.

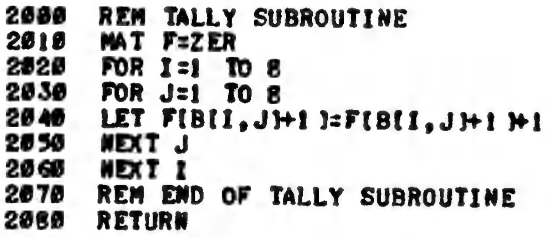


```
1000   REM FIGURE 5
1010   REM A QUANTUM SHUFFLING GAME
1020   DIM B[8,8],F[30]
1030   MAT  READ B
1040   REM CARRY OUT TALLY
1050   GOSUB 2000
1100   REM RANDOMIZE RANDOM NUMBER GENERATOR
1110   PRINT
1120   PRINT "INPUT ANY INTEGER BETWEEN 1 AND 100"
1130   INPUT A
1140   FOR I=1  TO A
1150   LET X=RND(0)
1160   NEXT I
1200   REM ENTER GAME LOOPS
1210   FOR M=1  TO 6
1220   REM PRINT BOARD AND TALLY
1230   GOSUB 3000
1240   FOR N=1  TO 500
1250   REM MAKE A MOVE
1260   GOSUB 4000
1270   NEXT N
1280   REM CARRY OUT TALLY
1290   GOSUB 2000
1300   NEXT M
1310   REM END OF GAME LOOPS
1320   STOP
  .
  .
  .
8001   DATA 1,1,1,1,1,1,1,1
8002   DATA 1,1,1,1,1,1,1,1
8003   DATA 1,1,1,1,1,1,1,1
8004   DATA 1,1,1,1,1,1,1,1
8005   DATA 1,1,1,1,1,1,1,1
8006   DATA 1,1,1,1,1,1,1,1
8007   DATA 1,1,1,1,1,1,1,1
8008   DATA 1,1,1,1,1,1,1,1
9999   END
```

Figure 5(a) — Program For Crystal Simulation

The array upon which the game will be played is called B and is dimensioned in line 1020. F, the array which will be used to contain the frequency distribution of quanta, is also set up here. The initial distribution of quanta on the array B is read in from the DATA statements in line 1030. Note that to change the initial distribution of quanta we have only to modify the numbers in the DATA statements in lines 8001 through 8008.

The TALLY subroutine branched to in line 1050 is contained in Figure 5(b). The purpose of this is to generate a frequency distribution of the number of squares containing 0, 1, 2, • • • quanta. This is done by using the number of quanta on each square as a subscript to increment an element of the array F. One is added to the subscript so that we can handle 0 as a subscript in the array.

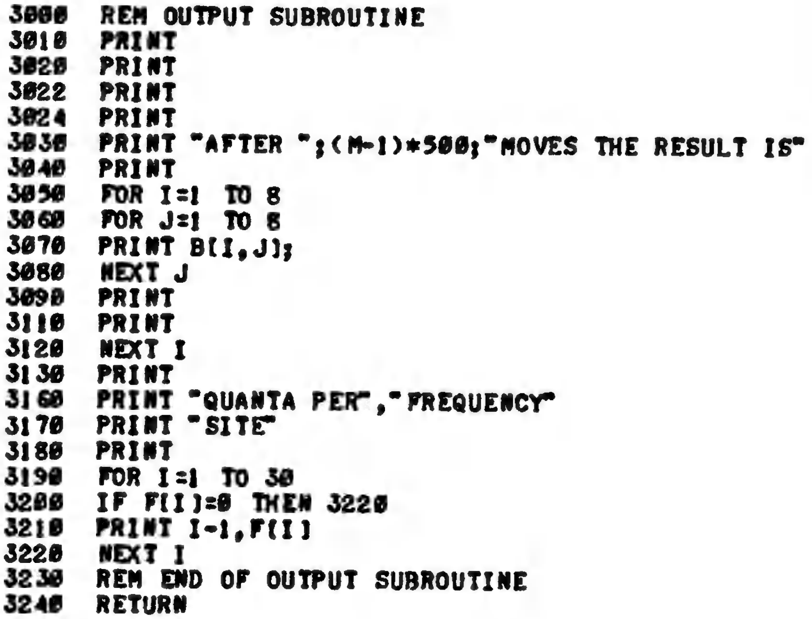


```
2ØØØ  REM TALLY SUBROUTINE
2Ø1Ø  MAT F=ZER
2Ø2Ø  FOR I=1  TO 8
2Ø3Ø  FOR J=1  TO 8
2Ø4Ø  LET F[B[I,J]+1 ]=F[B[I,J]+1 ]+1
2Ø5Ø  NEXT J
2Ø6Ø  NEXT I
2Ø7Ø  REM END OF TALLY SUBROUTINE
2Ø8Ø  RETURN
```

Figure 5(b) – TALLY Subroutine

The main loop to control the game is opened in line 1210 and closed in line 1300. The strategy is to print out the array and frequency distribution of quanta, make 500 moves, then repeat this cycle 5 times. The OUTPUT subroutine is in Figure 5(c) and can be followed without difficulty.

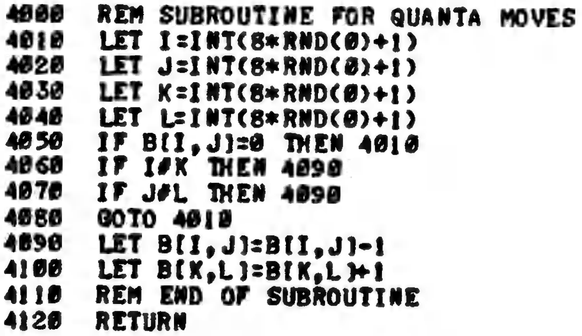
Note: *Lines 1100 through 1160 are needed only for HP Educational BASIC and 9830 systems.*



```
3ØØØ  REM OUTPUT SUBROUTINE
3Ø1Ø  PRINT
3Ø2Ø  PRINT
3Ø22  PRINT
3Ø24  PRINT
3Ø3Ø  PRINT "AFTER ";(M-1)*5ØØ;"MOVES THE RESULT IS"
3Ø4Ø  PRINT
3Ø5Ø  FOR I=1  TO 8
3Ø6Ø  FOR J=1  TO 8
3Ø7Ø  PRINT B[I,J];
3Ø8Ø  NEXT J
3Ø9Ø  PRINT
311Ø  PRINT
312Ø  NEXT I
313Ø  PRINT
316Ø  PRINT "QUANTA PER","FREQUENCY"
317Ø  PRINT "SITE"
318Ø  PRINT
319Ø  FOR I=1  TO 3Ø
32ØØ  IF F[I]=Ø  THEN 322Ø
321Ø  PRINT I-1,F[I]
322Ø  NEXT I
323Ø  REM END OF OUTPUT SUBROUTINE
324Ø  RETURN
```

Figure 5(c) – OUTPUT Subroutine

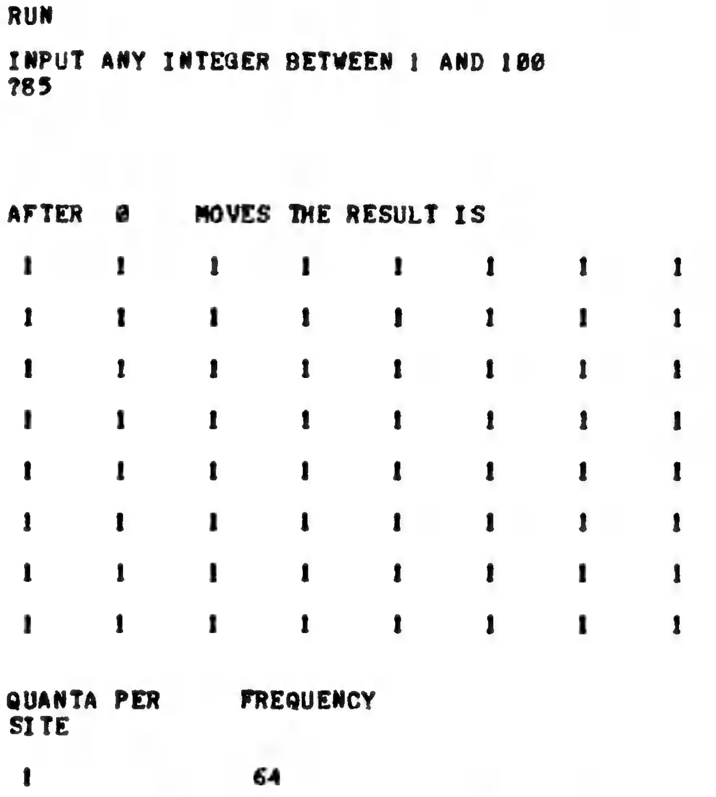
The heart of the game is the subroutine for moving quanta which is given in Figure 5(d).



```
4ØØØ  REM SUBROUTINE FOR QUANTA MOVES
4Ø1Ø  LET I=INT(8*RND(Ø)+1)
4Ø2Ø  LET J=INT(8*RND(Ø)+1)
4Ø3Ø  LET K=INT(8*RND(Ø)+1)
4Ø4Ø  LET L=INT(8*RND(Ø)+1)
4Ø5Ø  IF B[I,J]=Ø  THEN 4Ø1Ø
4Ø6Ø  IF I#K  THEN 4Ø9Ø
4Ø7Ø  IF J#L  THEN 4Ø9Ø
4Ø8Ø  GOTO 4Ø1Ø
4Ø9Ø  LET B[I,J]=B[I,J]-1
41ØØ  LET B[K,L]=B[K,L]+1
411Ø  REM END OF SUBROUTINE
412Ø  RETURN
```

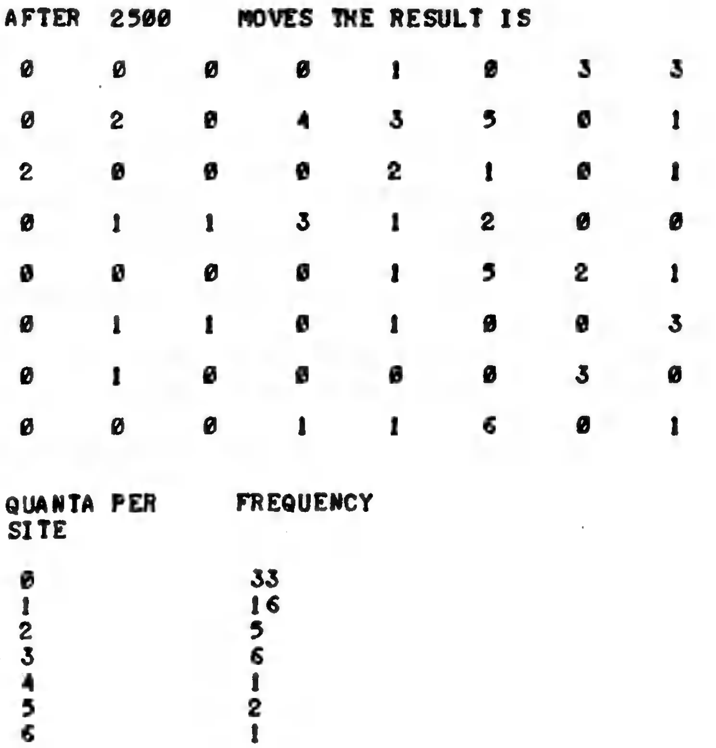
Figure 5(d) – MOVES Subroutine

You should study this subroutine carefully to see that it in fact follows the rules for the game which have been previously set down. Typical output from the computer game is shown in Figure 6. Only the first and last printouts are given. The first printout gives the initial distribution of quanta. The second gives the distribution after 2500 moves.



```
RUN

INPUT ANY INTEGER BETWEEN 1 AND 1ØØ
?85



AFTER  Ø    MOVES THE RESULT IS

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

 1     1     1     1     1     1     1     1

QUANTA PER     FREQUENCY
SITE

 1              64
```

Figure 6(a) — Initial Printout of Crystal Simulation Program



```
AFTER  25ØØ    MOVES THE RESULT IS

 Ø     Ø     Ø     Ø     1     Ø     3     3

 Ø     2     Ø     4     3     5     Ø     1

 2     Ø     Ø     Ø     2     1     Ø     1

 Ø     1     1     3     1     2     Ø     Ø

 Ø     Ø     Ø     Ø     1     5     2     1

 Ø     1     1     Ø     1     Ø     Ø     3

 Ø     1     Ø     Ø     Ø     Ø     3     Ø

 Ø     Ø     Ø     1     1     6     Ø     1

QUANTA PER     FREQUENCY
SITE

 Ø              33
 1              16
 2              5
 3              6
 4              1
 5              2
 6              1
```

Figure 6(b) — Final Printout of Crystal Simulation Program

Note carefully that the printout will be different for each initialization of the random number generator. (See note on page 17.) Therefore, you will get different numbers each time the program is run. We can, however, see something important in Figure 6. It is clear that the final distribution of quanta after 2500 moves is *completely* different from the initial distribution. Also, we should note that the only rule for moving quanta is blind chance.

EXERCISE 11 – A Computer Game

Enter the program in Figure 5 into your computer and run it. Examine the results carefully to see what is taking place. Does the initial distribution of quanta seem to be a reasonable one? Suppose the game were to be put through several more cycles. What changes do you believe would take place in the output?

INITIAL AND FINAL ENERGY DISTRIBUTION

The results of Exercise 11 illustrate several characteristics. First, a dramatic change in the distribution of energy has taken place. We started with 1 quantum of energy at each site, and finished with something quite different! The last printouts indicate that quanta are moving about on the array but the *distribution* of quanta seems to be remaining about the same. This distribution is such that it is most likely to find 0 quanta at a given site, and least likely to find many quanta at a given site. The frequency distribution of quanta per site versus number of sites appears to be a decreasing function of some type. Later we will look specifically at the form of this function.

A very important question remains unanswered. Does our initial distribution of quanta determine the final distribution? If we change the initial distribution will we get a new final distribution?

The computer permits us to investigate this easily. Recall that we have only to modify the numbers in the DATA statements in Figure 5(a) to change the initial distribution. Let us agree for the time being to always distribute 64 quanta over the 64 sites. Later we will relax this restriction. At this point, we are interested solely in investigating the effect of the initial distribution of quanta upon the final distribution. If you desire to suppress the printout of the array, insert the following statement in the program

```
3035 GO TO 3090
```

EXERCISE 12 – Energy Distribution 1

Run the program in Figure 5 with DATA statements modified to put 2 quanta on every other site. Compare the results with those from Exercise 11.

EXERCISE 13 – Energy Distribution 2

Repeat Exercise 12 except put 8 quanta on each site along one edge of the array.

EXERCISE 14 – Energy Distribution 3

Repeat Exercise 12 except put 16 quanta on each of the four corner sites in the array.

EXERCISE 15 – Arbitrary Distribution

Repeat Exercise 12 with a distribution of quanta of your choice. If you put a large number of quanta on one oscillator you may have to change the DIM statement in line 1020 of Figure 5(a).

DISTRIBUTION OF ENERGY

The computer results thus far point up a very important characteristic of our simulated crystal structure. No matter how the quanta are distributed initially, we *always* obtain the same distribution of quanta per site after a very large number of moves. There is some question about just what this distribution is since the computer results do not settle out on some fixed set of values but instead tend to oscillate statistically. Our task in this section is to learn more about this unknown distribution.

Suppose that the computer program to simulate random quanta interchange has been running for some time. We extract one frequency distribution of quanta per site and suppose it turns out to be that shown in Figure 7.

Quanta Per Site	Frequency
0	32
1	17
2	7
3	5
4	2
5	1
6	0

Figure 7 – Typical Distribution of Quanta In Simulated Crystal

Our experience with the computer simulation tells us that the outputs made both prior to and after the printout in Figure 7 will differ from that above. What we want to find is that distribution about which the individual distributions are scattered. If we were to take *many* individual distributions and average them, reason tells us that the result should be close to the theoretical distribution.

In any event, the distribution in Figure 7 points out rather clearly the type of mathematical function needed. A decreasing exponential would be an ideal function. Let us *assume* that the distribution is given by

$$n_j = n_0 e^{-j\beta\epsilon} , \tag{15}$$

where n_0 and β are unknown constants, j is an integer giving the number of quanta per site, and ϵ is the energy per quantum. In our exercises thus far, we have tacitly assumed that $\epsilon = 1$, although there is no need for this assumption. As we have set up the simulation, we generally will know N, M, and ϵ. N is the total number of sites, M is the number of quanta to be placed on the sites, and ϵ is the energy of each quanta. The total energy on the simulated crystal is given by

$$E_t = M\epsilon \ . \tag{16}$$

With this information, we want to find n_0 and β which will enable us to use (15) to compute the theoretical distribution.

To begin, we note that

$$N = \Sigma\, n_j \ , \tag{17}$$

or

$$\begin{aligned} N &= n_0 + n_0 e^{-\beta\epsilon} + n_0 e^{-2\beta\epsilon} + \cdots \\ &= n_0 \ (1 + e^{-\beta\epsilon} + e^{-2\beta\epsilon} + \cdots) \end{aligned} \tag{18}$$

The binomial theorem states that

$$(1 + e^{-\beta\epsilon} + e^{-2\beta\epsilon} + \cdots) = (1 - e^{-\beta\epsilon})^{-1} \tag{19}$$

provided that

$$(e^{-\beta\epsilon})^2 < 1 \ ,$$

which is true provided that both β and ϵ are greater than zero. We will assume this is the case.

Thus

$$N = n_0 (1 - \ e^{-\beta\epsilon})^{-1} \ . \tag{20}$$

The total energy of the system is given by

$$E_t = (n_0)\ (0) + (n_1)\ (\epsilon) + (n_2)\ (2\epsilon) + \cdots \tag{21}$$

But

$$n_j = n_0 e^{-j\beta\epsilon} \ ,$$

which leads to

$$\begin{aligned} E_t &= n_0 e^{-\beta\epsilon}\epsilon + n_0 e^{-2\beta\epsilon} 2\epsilon + \cdots \\ &= n_0 \epsilon e^{-\beta\epsilon} \ (1 + 2e^{-\beta\epsilon} + 3e^{-2\beta\epsilon} + \cdots) \end{aligned} \tag{22}$$

Again, the binomial theorem can be used to simplify the quantity inside parentheses on the right. The result is

$$E_t = n_0 \epsilon e^{-\beta\epsilon} (1 - e^{-\beta\epsilon})^{-2} \ . \tag{23}$$

The average energy per oscillator is E_t/N, or

$$\bar{E} = \frac{\epsilon}{e^{\beta\epsilon} - 1} \ . \tag{24}$$

Solving for β we obtain

$$\beta = \frac{1}{\epsilon} \ln(1 + \epsilon/\bar{E}) \ . \tag{25}$$

If we solve (20) for n_0 we have

$$n_0 = N \, (1 - e^{-\beta\epsilon}) \ . \tag{26}$$

Equations (25) and (26) enable us to derive a theoretical distribution given the energy per quantum ϵ, the total number of quanta M, and the total number of oscillators N. For the exercises we have done thus far, $\epsilon = 1$, $N = 64$, and $M = 64$. Consequently, $\bar{E} = 1$. Equations (25) and (26) yield $\beta = \ln(2)$, and $n_0 = 32$. The distribution is therefore

$$n_j = 32 \, e^{-j \ln(2)} = 32 \left(\frac{1}{2}\right)^j$$

Figure 8 shows the comparison between our derived distribution and the experimental one from Figure 7.

j	n_j	
	Experimental	Derived
0	32	32
1	17	16
2	7	8
3	5	4
4	2	2
5	1	1
6	0	0.5
7	0	0.25
8	0	0.125
9	0	0.0625
10	0	0.03125

Figure 8 – Comparison of Typical Computer Results and Derived Distribution

We have obtained a distribution of quanta per site which agrees well with Figure 7, and reasonably well with other computer results. Note carefully, however, that everything in the development flows from the assumption that

$$n_j = n_0 e^{-j\beta\epsilon} \tag{27}$$

The fact that (27) is the correct distribution function has not been proven. However, it turns out that it is the correct relation. The derivation and proof are given in most statistical physics texts.

Before going on, we should pause to consider the restrictions on the determination of β and n_0 by Equations (25) and (26). First, we used a binomial series to simplify expressions in two cases. This implies an infinite number of terms in the series. Actually, we have only from 6 to 10 terms in the actual computer results. Therefore, the fewer the number of terms, the greater the disagreement between theoretical and computer results. Also, in the development of (20) we require that β and ϵ be greater than zero. This has more significance than would appear at first glance and will be discussed later. For the present, we assume only that β and ϵ are positive constants.

The last item to be discussed in this section is the connection between the distribution function given by (27) and the probability of observing a site with j quanta. From (1),

$$P(j) = \frac{n_j}{N} = \frac{n_0}{N} e^{-j\beta\epsilon} \ .$$

However, $j\epsilon = E_j$ the energy on an oscillator with j quanta. Thus

$$P(j) = Ce^{-\beta E_j} \ , \tag{28}$$

where C is a constant (explained later in this discussion). The probability relation given by (28) is much more general than we might believe. In our treatment, j is an integer. More generally, if r is a specified state and E_r is the energy associated with that state, then

$$P(r) = Ce^{-\beta E_r} \ . \tag{29}$$

This expression has fundamental importance in statistical mechanics. The factor $e^{-\beta E_r}$ is known as the *Boltzmann factor.* The entire distribution given by (29) is called the *canonical distribution.* The constant C can be determined (in principle at least) by the condition that

$$\sum_r P(r) = 1 \ .$$

Thus

$$C = (\sum_r e^{-\beta E_r})^{-1} \tag{30}$$

EXERCISE 16 – An Average of Distributions

Modify the crystal simulation program to average 20 individual distributions of quanta per site. Start the program with DATA statements which *reflect a typical state had the program been running for some time. Compare the average of the frequency distributions to the theoretical distribution. Can you make any generalizations?*

EXERCISE 17 – A Larger Simulation

Modify the crystal simulation program to handle an N by N array with M quanta distributed on the array. Suppress the printout of the array itself. Output only the frequency distribution of quanta per site. Test your program on a 20 by 20 array containing 400 quanta. Compute the theoretical distribution and compare to your computer results.

EXERCISE 18 – Effect of Total Energy

Use the program from Exercise 17 with a 10 by 10 array. Run the program with 100, 200, 400, and 1000 quanta. Examine your results carefully, and in each case compare to the corresponding theoretical distribution.

NEW RULES – NEW DISTRIBUTIONS?

We have already investigated thoroughly the idea that the initial distribution of quanta has no effect upon the final distribution obtained after randomly moving quanta many times. In all cases, the exponential distribution obtained is quite independent of the initial distribution.

An equally important question has not been answered. Is it possible that the rules for moving quanta determine the final distribution? Certainly, if we think carefully about the simulation there are several aspects which might seem unreasonable and could suggest new rules. First, we have assumed that any oscillator can transfer a quantum of energy to any other oscillator in the array. How does this take place? We have avoided any discussion of the mechanism by which energy is transferred. However, whatever the mechanism is, it would seem reasonable that a quantum should be transferred to an adjacent site with a higher probability than a more distant site. Also, we have been transferring energy in discrete amounts. What would happen if we transferred fractional parts of the energy on a given oscillator?

Remember that the "rules" of the game are contained in the subroutine in Figure 5(d). If we change the rules, this is the primary point in the program that must be modified. If non-integer amounts of energy are transferred, then the amount of energy on each oscillator will, in general, be a non-integer value. We can still use the TALLY subroutine in Figure 5(b). However, BASIC will round non-integer values to the nearest integer when used as a subscript.

EXERCISE 19 – New Rules 1

Modify the rules of the crystal simulation as follows: If the donor oscillator is not on the edges of the array, select the oscillator to receive energy at random from the four closest neighbors. If the donor oscillator is on the edges of the array, but not one of the four corners, select the *oscillator to receive energy at random from the three closest neighbors. If the donor oscillator is a corner oscillator, select the oscillator to receive energy at random from the two closest neighbors. With these new rules, run the simulation for various-sized arrays and total number of quanta. In each case, compare the computer results with the Boltzmann distribution given by (27). Did the rule change affect the final distribution?*

EXERCISE 20 — New Rules 2

Modify the rules of the crystal simulation as follows: When the donor oscillator is selected, instead of transferring a quantum of energy, transfer a fractional part of the energy on the oscillator. Determine randomly what this fraction will be. Run the program for various-sized arrays and initial amounts of energy. In each case, compare the computer results with the Boltzman distribution given by (27). Did the rule change affect the final distribution?

EXERCISE 21 — Goodness Of Fit

One of the difficulties with Exercises 19 and 20 is how to detect if there is a significant difference between two distributions. Consult an introductory statistics text and learn how to perform a Chi-Squared test to accomplish this.

CHAPTER FOUR: TEMPERATURE AND HEAT FLOW

In the crystal simulation which has been the central part of our investigation so far, we have not mentioned two important terms. The first is *temperature,* which is commonly used to describe macrosystems. The second is *heat,* which has central importance in thermodynamics. We can look at both these concepts using our crystal simulation.

DEFINITION OF TEMPERATURE

Recall that the Boltzmann distribution is

$$n_j = n_0 e^{-j\beta\epsilon} \;, \qquad (31)$$

and that the average energy per oscillator is

$$\bar{E} = \frac{\epsilon}{e^{\beta\epsilon} - 1} \;. \qquad (32)$$

Thus far we have only defined β as a positive constant (for a given system) and have not associated it with any macroparameter. Suppose we compare two systems that are identical except one has a higher average energy per oscillator than the second:

$$\bar{E}_1 = \frac{\epsilon}{e^{\beta_1\epsilon} - 1} \;, \qquad (33)$$

$$\bar{E}_2 = \frac{\epsilon}{e^{\beta_2\epsilon} - 1} \;. \qquad (34)$$

If $\bar{E}_1 > \bar{E}_2$ then it follows from (33) and (34) that $\beta_2 > \beta_1$. In other words, as $\bar{E}$ increases, β decreases. Increasing the average energy per oscillator in the system corresponds to increasing the temperature of the system. This in turn must correspond to a decrease in β. It turns out that

$$\beta \propto \frac{1}{T} \;. \qquad (35)$$

The constant of proportionality in (35) is l/k where k is the *Boltzmann constant.* Therefore

$$\beta = \frac{1}{kT} \;, \qquad (36)$$

which connects temperature to the Boltzmann factor and the canonical distribution in (29). β must have dimensions of energy^{-1} and this is indeed so by (36). Also recall we have specified that β must always be positive. Since k is a positive physical constant, then T must always be positive. Thus T as determined by (36) is the *absolute temperature.*

Since we can compute β from information about the crystal simulation we can, in turn, compute the temperature of the system,

$$T = \frac{\epsilon}{k \ln(1 + \epsilon/\bar{E})} \quad . \qquad (37)$$

However, we must be careful about using (37) to compute the temperature. The relationship is valid *only* when the system is described by the Boltzmann distribution. If, for example, all the quanta are on the same oscillator, $\bar{E}$ is the same as when the quanta are distributed according to the Boltzmann distribution. However, the two situations are quite different. When the system is in *equilibrium* it is described by the Boltzmann distribution and the temperature can be determined from (37).

To see the effect of temperature, suppose we have a system composed of elements that are known to be in one of two energy levels. These energies are 0 and ϵ respectively. By (29)

$$P(0) = Ce^{-\beta(0)} \quad ,$$

and

$$P(\epsilon) = Ce^{-\beta\epsilon} \quad .$$

The constant C is determined by (30)

$$C = \left(e^{-\beta(0)} + e^{-\beta\epsilon}\right)^{-1} = \left(1 + e^{-\beta\epsilon}\right)^{-1} \quad .$$

Thus

$$P(0) = \left(1 + e^{-\beta\epsilon}\right)^{-1} \quad , \qquad (38)$$

and

$$P(\epsilon) = e^{-\beta\epsilon}\left(1 + e^{-\beta\epsilon}\right)^{-1} \quad . \qquad (39)$$

We can compute the average energy of each element in the system by the method described by (8).

$$\bar{E} = (0)P(0) + (\epsilon)P(\epsilon) \quad ,$$

or

$$\bar{E} = \frac{\epsilon e^{-\beta\epsilon}}{1 + e^{-\beta\epsilon}} \quad . \qquad (40)$$

It is a simple mathematical exercise to show that if the temperature approaches absolute zero, then $kT \rightarrow 0$, and $\bar{E} \rightarrow 0$. If the temperature increases without limit, $kT \rightarrow \infty$, and $\bar{E} \rightarrow \epsilon/2$. If there are N elements in the system, then the total energy of the system is simply N $\bar{E}$.

With this simple case there is no need to involve the computer. However, with more involved (and more interesting) cases where the energy level structure is more complicated, the computations are much more involved. Applications of the computer to this type of problem have been described by Weinstock.[5]

EXERCISE 22 – Unequally-Spaced Three Level System

A system is composed of units each of which has the permissible energy levels below. In each case, write a program to compute the average energy per unit versus kT/ϵ. Consider at least the range $0.05 < kT/\epsilon < 3.0$. Plot the results.

a) $0, \epsilon, 3\epsilon$

b) $0, 2\epsilon, 3\epsilon$

c) $0, \epsilon, 5\epsilon$

d) $0, 4\epsilon, 5\epsilon$

EXERCISE 23 – Equally-Spaced Level System

The harmonic oscillators of our crystal simulation have equally spaced energy levels: $0, \epsilon, 2\epsilon, 3\epsilon, \cdots$. Find the average energy per oscillator versus kT/ϵ for the cases below. Consider at least the range $0.05 < kT/\epsilon < 3.0$.

a) $0, \epsilon, 2\epsilon$

b) $0, \epsilon, 2\epsilon, \cdots, 10\epsilon$

c) $0, \epsilon, 2\epsilon, \cdots, 30\epsilon$

EXERCISE 24 – Inverse Square-Spaced Level System

Some systems have energy levels of the form

$$E_n = -\frac{C}{n^2} \qquad n = 1, 2, 3 \cdots$$

In our analysis we want to have the lowest energy level be 0. The expression above modified to do this, and, with an assumed value of C, is

$$E_n = 5\epsilon \left(1 - \frac{1}{n^2}\right) \qquad n = 1, 2, 3, \cdots$$

Write a program to find the average energy per unit versus kT/ϵ for such a system. Use $n = 1, 2, 3, \cdots, 10$ and a reasonable range of kT/ϵ. Plot the results.

EXERCISE 25 – Randomly-Spaced Level System

One of the interesting things about investigations with the computer is that we can simulate any type of universe we desire. It makes no difference whether there is a real counterpoint to the simulation or not. Suppose that natural law was such that a system had units which had randomly-spaced energy levels. How would the average energy per unit versus kT/ϵ behave for such a system? Assume that five energy levels are selected at random from the range 0 to 5ϵ.

HEAT CAPACITY

With the foundation established in the previous section, we can very quickly bring out the concept of *heat capacity*. Suppose that a system has average energy per unit of $\bar{E}$. In the crystal simulation, the *unit* was an oscillator. In other systems, the unit might be something different. The essential idea is that the system is composed of units, each of which can hold energy, and has an average energy. The total energy of the system is the sum of the average values of all the units in the system.

Exercises 22 through 25 were concerned with finding $\bar{E}$ as a function of kT/ϵ. In other words, we found how the average energy per unit depended upon the temperature of the system. Recall that for the very simple system in which the units could be only in the levels 0 or ϵ, the average energy per unit was

$$\bar{E} = \frac{\epsilon e^{-\beta\epsilon}}{1 + e^{-\beta\epsilon}} , \qquad (41)$$

where

$$\beta = \frac{1}{kT} .$$

The heat capacity (at constant volume) per unit is defined as the rate of change of $\bar{E}$ with respect to T,

$$C_v = (\partial\bar{E}/\partial T)_v . \qquad (42)$$

Since (41) depends explicitly on β, we can use the chain rule of differentiation to give

$$C_v = \left(\frac{\partial\bar{E}}{\partial\beta}\right)_v \left(\frac{d\beta}{dT}\right) . \qquad (43)$$

Carrying out this differentiation and simplifying we have the following expression for the heat capacity per unit for the system described by (41),

$$C_v = k(\beta\epsilon)^2 \frac{e^{-\beta\epsilon}}{(1 + e^{-\beta\epsilon})^2} . \qquad (44)$$

It will simplify our computer exercises if we compute C_v/k instead of C_v. Thus,

$$\frac{C_v}{k} = (\beta\epsilon)^2 \frac{e^{-\beta\epsilon}}{(1 + e^{-\beta\epsilon})^2} . \qquad (45)$$

For high temperatures, $\beta\epsilon << 1$ and C_v/k falls off as the inverse square of the temperature. For low temperatures, $\beta\epsilon >> 1$ and C_v/k rises exponentially.

The only reason we can write (45) is that the functional dependence of $\bar{E}$ upon β (and therefore T) is known. Generally this information will not be available to us. However, we *can* compute and tabulate $\bar{E}$ versus kt/ϵ. Using the computer we can then numerically compute the derivative of $\bar{E}$

with respect to T:

$$C_v = \frac{\partial \bar{E}}{\partial T} = \frac{\partial \bar{E}}{\partial (kT/\epsilon)} \frac{d(kT/\epsilon)}{dT} = \frac{k}{\epsilon} \frac{\partial \bar{E}}{\partial (kT/\epsilon)}$$

or,

$$\frac{\epsilon}{k} C_v = \frac{\partial \bar{E}}{\partial (kT/\epsilon)} \quad . \tag{46}$$

Now we need to compute numerically the right side of (46). Suppose that kT/ϵ takes on the values .05, .10, • • • , 2.95, 3.00. If i = 1, 2, 3, • • • , 60, then the values of kT/ϵ are simply .05i. We can compute $\bar{E}_i$ for each of the values of i. The first derivative of $\bar{E}$ with respect to kT/ϵ (using a central difference expression) is[2]:

$$\frac{\partial \bar{E}_i}{\partial (kT/\epsilon)} = \frac{\bar{E}_{i+1} - \bar{E}_{i-1}}{2\Delta(kT/\epsilon)} \quad .$$

But since $\Delta(kt/\epsilon) = .05$, we have

$$\left(\frac{C_v}{k}\right)_i = \frac{\bar{E}_{i+1} - \bar{E}_{i-1}}{0.1} \quad . \tag{47}$$

We have dropped the ϵ on the left since $\bar{E}$ is determined in terms of ϵ and they cancel. Using (47) we can compute C_v/k at each of the points in the array of kT/ϵ values to get a profile. Remember that we must multiply the computed values by k to get the heat capacity per unit, and multiply by Nk to get the heat capacity of the system, where N is the number of units in the system.

The primary purpose here is not to compute numerical values of heat capacity. Rather, it is to compute the draw profiles of how the heat capacity varies with temperature and thereby gain insight into the behavior of the systems.

EXERCISE 26 – Heat Capacity, Unequally-Spaced Three Level System

Write a program to compute the heat capacity per unit for each of the systems in Exercise 22. Plot your results.

EXERCISE 27 – Heat Capacity, Equally-Spaced Level System

Write a program to compute the heat capacity per unit for each of the systems in Exercise 23. Plot your results.

EXERCISE 28 – Heat Capacity, Inverse Square-Spaced Level System

Write a program to compute the heat capacity per unit for the system in Exercise 24. Plot your results.

EXERCISE 29 – Heat Capacity, Randomly Spaced Level System

Write a program to compute the heat capacity per unit for the system in

Exercise 25. Plot your results.

HEAT TRANSFER

The crystal model furnishes an ideal way to study heat transfer. First, we must be very specific in our definition of heat. Heat is energy transferred in the absence of any external work. In our crystal model, we could arbitrarily divide the system (the crystal) into two smaller subsystems. We could take these to be the left and right halves of the crystal respectively. Now, as quanta are transferred between the oscillators, energy is clearly flowing from one subsystem to the other. It is also clear that no external work is involved. As indicated above, the energy thus transferred is defined as heat.

Several important questions can be investigated using the simulated crystal model. Suppose two systems at the same temperature are brought together and allowed to come to equilibrium. Will the distribution of quanta differ from distributions in the two separated systems? What will the outcome be if the two different systems are initially at different temperatures? Finally, suppose a small system at equilibrium and at one temperature is brought into contact with a large system at equilibrium at another temperature. What will take place?

By this time you should be convinced that if we place M quanta on N oscillators and move them about randomly, after a sufficiently-long time the quanta will be distributed according to the Boltzmann distribution and we can compute the temperature using (37). In the exercises to follow, the computer is not needed. You can reach a solution to each of the exercises using your knowledge of how blind chance affects systems, the fact that $\bar{E}$ is the average energy per oscillator, and equation (37).

EXERCISE 30 – Two Systems, Same Temperature

Two systems at the same temperature are isolated from one another for a long time. They are then brought together and allowed to reach equilibrium. What will the temperature of the combined system be? Will any heat flow take place?

EXERCISE 31 – Two Systems, Different Temperature

System A is at equilibrium and has temperature T_A. System B is also at equilibrium and at temperature T_B. Suppose that $T_A > T_B$. If the two systems are brought into contact and permitted to exchange energy, what will the final temperature be? Will heat flow? If so, in which direction?

EXERCISE 32 – Two Systems: One Large, One Small

A small system at equilibrium is isolated from a large system also at equilibrium. Suppose that the temperature of the small system is not the same as that of the large one. If the two systems are brought together, describe what will take place.

CHAPTER FIVE: ENTROPY AND EQUILIBRIUM

ACCESSIBLE STATES AND ENTROPY

The introduction pointed out that the bridge between macrostates and microstates is furnished by the relation

$$S = k \ln(\Omega) \ , \tag{48}$$

where S is the *entropy* of a system, k is the Boltzmann constant and Ω is the number of microstates accessible to the system. The new concept here is that of accessible microstates. An example will illustrate the essential ideas: Suppose we have a system composed of three oscillators which share a total of 5 quanta, where each quantum of energy has magnitude ϵ. Figure 9 enumerates all possible combinations which are allowable.

Configuration	Microstate	Oscillator		
		A	B	C
1	1	5ϵ	0	0
	2	0	5ϵ	0
	3	0	0	5ϵ
2	4	4ϵ	ϵ	0
	5	4ϵ	0	ϵ
	6	ϵ	4ϵ	0
	7	0	4ϵ	ϵ
	8	ϵ	0	4ϵ
	9	0	ϵ	4ϵ
3	10	3ϵ	2ϵ	0
	11	3ϵ	0	2ϵ
	12	2ϵ	3ϵ	0
	13	0	3ϵ	2ϵ
	14	2ϵ	0	3ϵ
	15	0	2ϵ	3ϵ
4	16	3ϵ	ϵ	ϵ
	17	ϵ	3ϵ	ϵ
	18	ϵ	ϵ	3ϵ
5	19	2ϵ	2ϵ	ϵ
	20	2ϵ	ϵ	2ϵ
	21	ϵ	2ϵ	2ϵ

Figure 9 — Configurations and Microstates for Three Oscillators Sharing Five Quanta of Energy

There are five distinguishable configurations for the system, and a total of 21 allowed microstates. The configurations are determined by the frequency distribution of quanta per site. These distributions are shown in Figure 10.

Configuration	Quanta Per Site	Frequency
1	0	2
	5	1
2	0	1
	1	1
	4	1
3	0	1
	2	1
	3	1
4	1	2
	3	1
5	1	1
	2	2

Figure 10 — Distribution of Quanta for Possible Configurations

The frequency distribution of quanta per site identifies each of the possible configurations. This is just the familiar frequency distributions which were output from the crystal simulation. We can use this information to calculate the number of microstates corresponding to each configuration:

$$\Omega = \frac{N!}{(f_1!)\,(f_2!)\,(f_3!)\cdots} \ . \tag{49}$$

In (49), N is the number of oscillators and f_1, f_2, f_3, $\cdots$ are the frequencies in the distribution of quanta per site. From Figure 10 we have

$$\Omega_1 = \frac{3!}{(2!)\,(1!)} = 3 \ ,$$

$$\Omega_2 = \frac{3!}{(1!)\,(1!)\,(1!)} = 6 \ ,$$

$$\Omega_3 = \frac{3!}{(1!)\,(1!)\,(1!)} = 6 \ ,$$

$$\Omega_4 = \frac{3!}{(2!)\,(1!)} = 3 \ ,$$

$$\Omega_5 = \frac{3!}{(1!)\,(2!)} = 3 \ .$$

These computations agree exactly with the results enumerated in Figure 9. Using (49) we can compute Ω for any configuration. However, rather than computing Ω, we need $\ln(\Omega)$ since this is required to compute the entropy

S. Taking the natural logarithm of both sides of (49) we have

$$\ln(\Omega) = \ln(N!) - \ln(f_1!) - \ln(f_2!) - \ln(f_3!) \cdots \qquad (50)$$

It is not possible to evaluate the factorials in (50) on the computer and then take the required logarithms, because of the size of factorials. If, as in our first crystal simulation, N = 64, then $N! = 1.2689 \times 10^{89}$. This very large number is outside the range of many computers. We can solve the problem by using Stirling's approximation for n!.

$$\ln(n!) = \left(n + \frac{1}{2}\right)\ln(n) - n + \ln\sqrt{2\pi}\ ,$$

which if the numerical value for $\ln\sqrt{2\pi}$ is inserted becomes

$$\ln(n!) = \left(n + \frac{1}{2}\right)\ln(n) - n + 0.918939\ . \qquad (51)$$

The approximation gets better as n increases. For n = 5 the error is about 16 parts in ten thousand. For n = 50, the error is about 2 parts in a million. The error for small n does not play an important part in the computer investigation of entropy. Consequently we will always employ Stirling's approximation given by (51) to compute the logarithms of the factorials required.

It will be simpler to solve for S/k rather than S. Thus

$$\frac{S}{k} = \ln(N!) - \ln(f_1!) - \ln(f_2!) - \cdots \qquad (52)$$

where each of the terms on the right is computed using (51).

EXERCISE 33 – Entropy Calculation 1

Modify the program for the crystal simulation to produce a printout of S/k for each of the frequency distributions as they are computed. Set up a 10 by 10 array which contains

a) 200 quanta

b) 100 quanta

Examine the results carefully and try to generalize what is taking place. Experiment with various numbers of moves between printouts of S/k.

EXERCISE 34 – Entropy Calculation 2

Repeat Exercise 33 except use a 5 by 5 array which contains

a) 50 quanta

b) 25 quanta

EQUILIBRIUM

We are now in a position to firm up previous discussions about equilibrium. First, we must return to the information in Figure 9 to make a fundamental point concerning statistical mechanics. We had 3 oscillators sharing 5 quanta of energy which gave rise to 21 allowable microstates. The key postulate of statistical mechanics is that **all microstates are equally likely**. Thus, configuration 1, which contains 3 of the possible 21 microstates, will be observed with a probability of 3/21. Likewise, the probability of observing the system in configuration 3 is 6/21, and so on. It follows that the most probable configuration is the one corresponding to the largest number of microstates.

Suppose we look at a typical example to make this point clear. Two frequency distributions of quanta per site are given below. In each case, we will assume N = 64, and 64 quanta.

Case 1		Case 2	
Quanta Per Site	Frequency	Quanta Per Site	Frequency
0	63	0	32
64	1	1	16
		2	8
		3	5
		4	3
		5	1

From (49), we have

$$\Omega_1 = \frac{64!}{(63!)\,(1!)} = 64, \text{ and}$$

$$\Omega_2 = \frac{64!}{(32!)\,(16!)\,(8!)\,(5!)\,(3!)\,(1!)} = 7.932 \times 10^{32} .$$

Therefore, the probability of configuration 2 is about 10^{31} greater than the probability of configuration 1. The system will be found with overwhelming probability in that configuration which corresponds to the greatest number of microstates. It is possible but not probable to find a system in a configuration with a small number of microstates. Just how small this probability is can be seen in the computations for Ω_1 and Ω_2 above.

The equilibrium configuration of a system is that one which corresponds to the greatest number of microstates. The Boltzmann distribution of quanta per site corresponds to a greater number of microstates than any other distribution. If a system is permitted to respond to blind chance in the interchange of quanta, it will drift towards the equilibrium configuration which implies the largest number of microstates.

Since S/k is the natural logarithm of the number of accessible states, we can immediately conclude that as a system goes from non-equilibrium toward equilibrium, the entropy will increase. Moreover, the equilibrium configuration corresponds to a maximum in entropy for the system.

EXERCISE 35 – Entropy and the Boltzmann Distribution

Figure 8 contains the Boltzmann distribution for 64 quanta on 64 oscillators. The first 6 terms are: 32, 16, 8, 4, 2, and 1. These terms add up to 57 quanta on 63 oscillators because of truncation error. Write a program to compute S/k. Use this program to investigate the effect of changing the distribution (keeping 57 quanta and 63 oscillators). Compare to the Boltzmann distribution.

CHAPTER SIX: FINAL THOUGHTS

There have been a number of assumptions made throughout this unit. As long as the subject is restricted to classical statistical mechanics, it is not necessary to raise the issue. However, as soon as we consider quantum statistics, it is important to understand just what these assumptions are.

We have assumed that the oscillators of our simulated crystal were identical in all respects save that of location. We could identify each oscillator by position and determine the number of quanta of energy held by that oscillator. In other words, the system was composed of a number of *distinguishable* units. Such a system is described by Boltzmann statistics as we have seen.

If, however, we consider a system of particles which are not distinguishable, the situation is significantly different. An example might be a group of identical atoms moving inside a rigid box. The atoms move freely in a common potential well which is not divided up into local valleys (which would correspond to our oscillators in the crystal simulation). Such particles cannot be assigned to a location in the box. In principle, we can observe the energy of every particle but there is no way to determine which of the particles had which of the observed energy values. Thus the particles are *indistinguishable.*

If the set of indistinguishable particles obeys the exclusion principle (no two particles can be in the same dynamical state) then the system is described by *Fermi-Dirac* statistics. If the set of indistinguishable particles is not restricted by the exclusion principle, then the system is described by *Bose-Einstein* statistics. Quantum statistics is concerned with these two new types of statistics.

REFERENCES

(1) P.J. Black, "A Quantum Shuffling Game For Teaching Statistical Mechanics", *American Journal of Physical,* Vol 39, October 1971.

(2) H. Peckham, *Computers, BASIC, & Physics,* Addison-Wesley Inc., 1971, p. 55.

(3) F. Reif, *Statistical Physics,* Vol 5, Berkeley Physics Course, McGraw Hill, New York, 1967.

(4) C.W. Sherwin, *Basic Concepts Of Physics,* Holt, Rinehart & Winston, Inc., New York, 1961.

(5) H. Weinstock, *Statistical Physics Computer Applications,* Proceedings of the 1972 Conference on Computers in Undergraduate Curricula.

ANSWERS TO SELECTED EXERCISES

CHAPTER TWO

EXERCISE 1 – Flipping Coins

The theoretical results are P(0) = P(5) = 0.03125, P(1) = P(4) = 0.15625, and P(2) = P(3) = 0.31250. The program and results after 451 throws follow.

```
LIST
100   REM EXERCISE 1
110   DIM F[6]
120   MAT F=ZER
130   PRINT
140   LET C=0
150   FOR I=1  TO 500
160   IF C<50  THEN 240
170   PRINT "AFTER";I;"TOSSES THE RESULTS ARE"
180   PRINT "HEADS","PROBABILITY"
190   PRINT
200   FOR J=1  TO 6
210   PRINT J-1,F[J]/I
220   NEXT J
225   PRINT
226   PRINT
230   LET C=0
240   LET S=1
250   FOR K=1  TO 5
260   LET S=S+INT(RND(0)+.5)
270   NEXT K
280   LET F[S]=F[S]+1
290   LET C=C+1
300   NEXT I
999   END

RUN


AFTER 451   TOSSES THE RESULTS ARE
HEADS          PROBABILITY

 0              3.10421 E-02
 1              .195122
 2              .277162
 3              .330377
 4              .13969
 5              2.43902 E-02
```

EXERCISE 3 – A Data Set

```
100   REM EXERCISE 3
110   LET N=S1=S2=0
120   READ X
130   IF X=9999  THEN 180
140   LET N=N+1
150   LET S1=S1+X
160   LET S2=S2+X↑2
170   GOTO 120
180   PRINT
190   PRINT "MEAN = ";S1/N
200   PRINT "VARIANCE = ";(N*S2-S1↑2)/N↑2
800   DATA 9.1,10.3,9.8,9.9,10.1,10,10.2,9999
999   END
RUN

MEAN =  9.91429
VARIANCE =  .135543
```

EXERCISE 5 – A Probability Distribution

```
100  REM EXERCISE 5
110  LET S1=S2=0
120  READ N
130  FOR I=1 TO N
140  READ X,P
150  LET S1=S1+X*P
160  LET S2=S2+X↑2*P
170  NEXT I
180  LET M=S1
190  LET V=S2-S1↑2
200  PRINT " MEAN = ";M
210  PRINT " VARIANCE = ";V
800  DATA 9
801  DATA 1,4.00000E-02
802  DATA 2,8.00000E-02
803  DATA 3,.12
804  DATA 4,.16
805  DATA 5,.2
806  DATA 6,.16
807  DATA 7,.12
808  DATA 8,8.00000E-02
809  DATA 9,4.00000E-02
999  END

RUN
MEAN =  5
VARIANCE =  4
```

EXERCISE 7 – A Biased Random Walk

According to the results of three different starting points, the answer is between step 1 and 2.

```
100  REM EXERCISE 7
110  PRINT "INPUT STARTING POINT";
120  INPUT X1
130  LET L=R=0
140  FOR I=1 TO 100
145  LET X=X1
150  IF RND(0)<.6 THEN 190
170  LET X=X-1
180  GOTO 200
190  LET X=X+1
200  IF X<0 THEN 230
210  IF X>10 THEN 250
220  GOTO 150
230  LET L=L+1
240  GOTO 260
250  LET R=R+1
260  NEXT I
270  PRINT "P(L) = ";L/100
280  PRINT "P(R) = ";R/100
290  PRINT
300  GOTO 110
999  END

RUN
INPUT STARTING POINT?0
P(L) =  .62
P(R) =  .38

INPUT STARTING POINT?1
P(L) =  .41
P(R) =  .59

INPUT STARTING POINT?2
P(L) =  .21
P(R) =  .79
```

EXERCISE 9 — Relative Fluctuations

```
1ØØ   REM EXERCISE 9
11Ø   DIM B[1ØØ]
12Ø   MAT B=ZER
13Ø   PRINT
14Ø   PRINT "INPUT INTEGER BETWEEN 1 AND 1ØØ"
15Ø   INPUT Z
16Ø   FOR I=1  TO Z
17Ø   LET X=RND(Ø)
18Ø   NEXT I
19Ø   PRINT "INPUT TOTAL NUMBER OF PARTICLES"
2ØØ   PRINT "MUST BE EVEN AND NO MORE THAN 1ØØ"
21Ø   INPUT N
22Ø   LET L=R=N/2
23Ø   FOR I=1  TO L
24Ø   LET B[I]=1
25Ø   NEXT I
26Ø   PRINT
27Ø   LET S1=S2=Ø
28Ø   FOR I=1  TO 1ØØ
29Ø   LET S1=S1+L
3ØØ   LET S2=S2+L↑2
31Ø   LET K=INT(N*RND(Ø)+1)
32Ø   IF B[K]=1  THEN 38Ø
33Ø   LET B[K]=1
34Ø   LET L=L+1
36Ø   LET R=R-1
37Ø   GOTO 41Ø
38Ø   LET B[K]=Ø
39Ø   LET L=L-1
4ØØ   LET R=R+1
41Ø   NEXT I
42Ø   LET M=S1/1ØØ
43Ø   LET S=SQR((1ØØ*S2-S1*S1)/1ØØØØ)
44Ø   PRINT "RATIO OF STND DEV TO MEAN = ";S/M
999   END
```

```
RUN

INPUT INTEGER BETWEEN 1 AND 1ØØ
?67
INPUT TOTAL NUMBER OF PARTICLES
MUST BE EVEN AND NO MORE THAN 1ØØ
?4Ø

RATIO OF STND DEV TO MEAN =  .17Ø249

READY
RUN

INPUT INTEGER BETWEEN 1 AND 1ØØ
?3
INPUT TOTAL NUMBER OF PARTICLES
MUST BE EVEN AND NO MORE THAN 1ØØ
?6Ø

RATIO OF STND DEV TO MEAN =  .114215

READY
RUN

INPUT INTEGER BETWEEN 1 AND 1ØØ
?81
INPUT TOTAL NUMBER OF PARTICLES
MUST BE EVEN AND NO MORE THAN 1ØØ
?8Ø
```

EXERCISE 9 – (Continued)

```
RATIO OF STND DEV TO MEAN =  6.13737E-02

READY
RUN

INPUT INTEGER BETWEEN 1 AND 100
?43
INPUT TOTAL NUMBER OF PARTICLES
MUST BE EVEN AND NO MORE THAN 100
?100

RATIO OF STND DEV TO MEAN =  5.81937E-02
```

CHAPTER THREE

EXERCISE 11 – A Computer Game

The first and last printouts are contained in Figures 6(a) and 6(b) respectively. After 500 moves, the frequency distribution is oscillating about the equilibrium value. Results in this exercise and all others that utilize the random number generator will be different depending upon how the random number generator is sequenced in your computer. The trends should remain the same, however.

EXERCISE 13 – Energy Distribution 2

Modify the DATA statements in the program in Figure 5 as shown below. The distribution goes over to the same type as for Exercise 11.

```
8001   DATA 8,8,8,8,8,8,8,8
8002   DATA 0,0,0,0,0,0,0,0
8003   DATA 0,0,0,0,0,0,0,0
8004   DATA 0,0,0,0,0,0,0,0
8005   DATA 0,0,0,0,0,0,0,0
8006   DATA 0,0,0,0,0,0,0,0
8007   DATA 0,0,0,0,0,0,0,0
8008   DATA 0,0,0,0,0,0,0,0
9999   END
```

EXERCISE 15 – Arbitrary Distribution

The quanta can be distributed in any fashion desired on the oscillators. After a sufficiently large number of moves, all initial distributions will drift over to the type of distribution seen in Exercise 11.

EXERCISE 17 – A Larger Simulation

The complete program is given below. Depending upon the size of the program space available to you, the DIM statement may have to be modified. Also, if you have sufficient program space you can continue to specify the initial configuration with DATA statements. With limited space, other provisions have to be made. Following the program, the first few printouts are shown. The printouts are made every 50 moves to watch the process more closely.

```
1000   REM EXERCISE 17
1010   REM LARGER SIMULATION
1020   REM MAX SIZE IS 20 BY 20
1030   DIM B(20,20),F(30)
1040   PRINT "INPUT SIDE OF SQUARE ARRAY";
1050   INPUT N1
```

EXERCISE 17 – (Continued)

```
1060   MAT B=CON[N1,N1]
1070   REM CARRY OUT TALLY
1080   GOSUB 2000
1100   REM RANDOMIZE RANDOM NUMBER GENERATOR
1110   PRINT
1120   PRINT "INPUT ANY INTEGER BETWEEN 1 AND 100";
1130   INPUT A
1140   FOR I=1 TO A
1150   LET X=RND(0)
1160   NEXT I
1200   REM ENTER GAME LOOPS
1210   FOR M1=1 TO 111
1220   REM PRINT TALLY
1230   GOSUB 3000
1240   FOR N=1 TO 50
1250   REM MAKE  A MOVE
1260   GOSUB 4000
1270   NEXT N
1280   REM CARRY OUT TALLY
1290   GOSUB 2000
1300   NEXT M1
1310   REM END OF GAME LOOP
1320   STOP



2000   REM TELLY SUBROUTINE
2010   MAT F=ZER
2020   FOR I=1 TO N1
2030   FOR J=1 TO N1
2040   LET F[B[I,J]+1 ]=F[B[I,J]+1 ]+1
2050   NEXT J
2060   NEXT I
2070   REM END OF TALLY SUBROUTINE
2080   RETURN
3000   REM OUTPUT SUBROUTINE
3010   PRINT
3020   PRINT
3030   PRINT
3040   PRINT "AFTER";(M1-1)*50;"MOVES THE RESULT IS"
3050   PRINT
3060   PRINT "QUANTA PER","FREQUENCY"
3070   PRINT "SITE"
3080   PRINT
3090   FOR I=1 TO 30
3100   IF F[I]=0 THEN 3120
3110   PRINT I-1,F[I]
3120   NEXT I
3130   REM END OF OUTPUT SUBROUTINE
3140   RETURN
4000   REM SUBROUTINE FOR MOVES
4010   LET I=INT(N1*RND(0)+1)
4020   LET J=INT(N1*RND(0)+1)
4030   LET K=INT(N1*RND(0)+1)
4040   LET L=INT(N1*RND(0)+1)
4050   IF B[I,J]=0 THEN 4010
4060   IF I#K THEN 4090
4070   IF J#L THEN 4090
4080   GOTO 4010
4090   LET B[I,J]=B[I,J]-1
4100   LET B[K,L]=B[K,L]+1
4110   REM END OF SUBROUTINE
4120   RETURN
9999   END
```

EXERCISE 17 – (Continued)

```
RUN

INPUT SIDE OF SQUARE ARRAY?2Ø

INPUT ANY INTEGER BETWEEN 1 AND 1ØØ?2Ø


AFTER Ø     MOVES THE RESULT IS

QUANTA PER      FREQUENCY
SITE

 1               4ØØ


AFTER 5Ø    MOVES THE RESULT IS

QUANTA PER      FREQUENCY
SITE

 Ø               45
 1               312
 2               41
 3               2


AFTER 1ØØ   MOVES THE RESULT IS

QUANTA PER      FREQUENCY
SITE

 Ø               77
 1               253
 2               63
 3               7
```

A number of printouts have been omited. The final one is shown below.

```
AFTER 5ØØ   MOVES THE RESULT IS

QUANTA PER      FREQUENCY
SITE

 Ø               176
 1               1Ø8
 2               72
 3               3Ø
 4               12
 5               2
```

EXERCISE 19 – New Rules 1

This exercise requires careful thought to decide what moves are possible for each oscillator as it is selected. The logic should be worked out in flowchart form to insure that the algorithm is correct.

EXERCISE 21 – Goodness Of Fit

Let $f_1, f_2, f_3, \cdots$ be the observed frequencies.

Let $f'_1, f'_2, f'_3, \cdots$ be the expected frequencies according to the Boltzmann distribution.

Compute

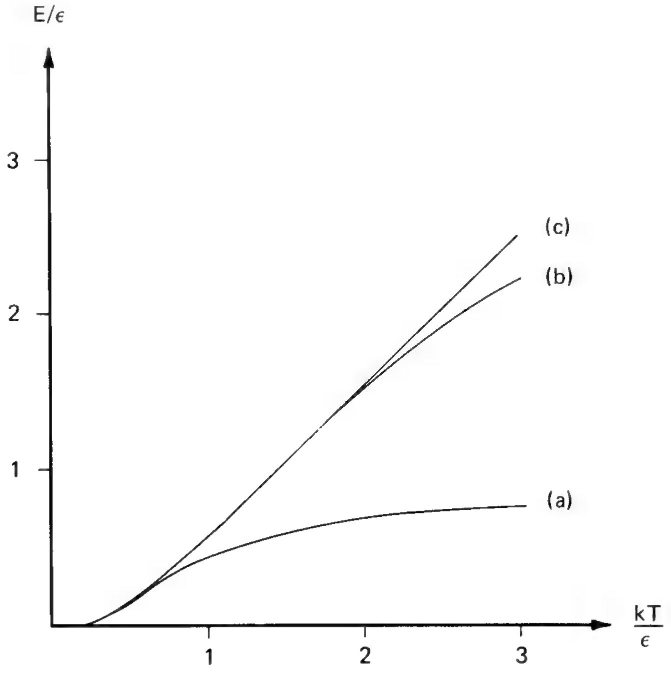
$$\chi^2 = \sum_i \frac{(f_i - f'_i)^2}{f'_i} ,$$

then proceed as described in statistics texts.

CHAPTER FOUR

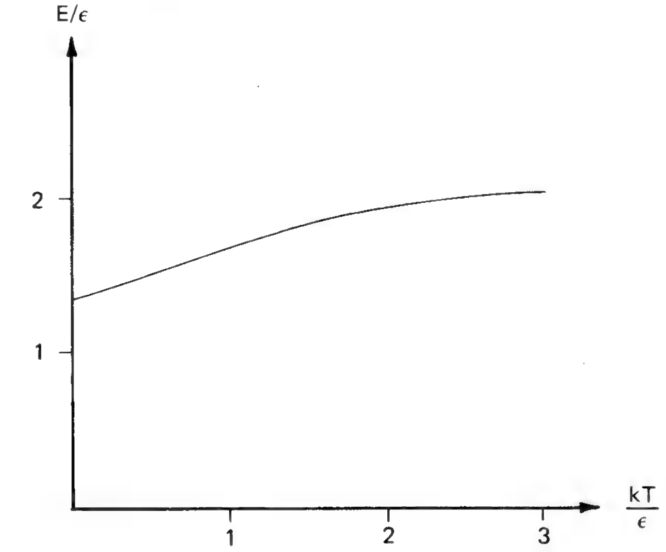
EXERCISE 23 – Equally-Spaced Level System

The results are graphed below. Note that for 30 equally-spaced energy levels, the graph is approaching the linear relationship for an infinite number of energy levels.



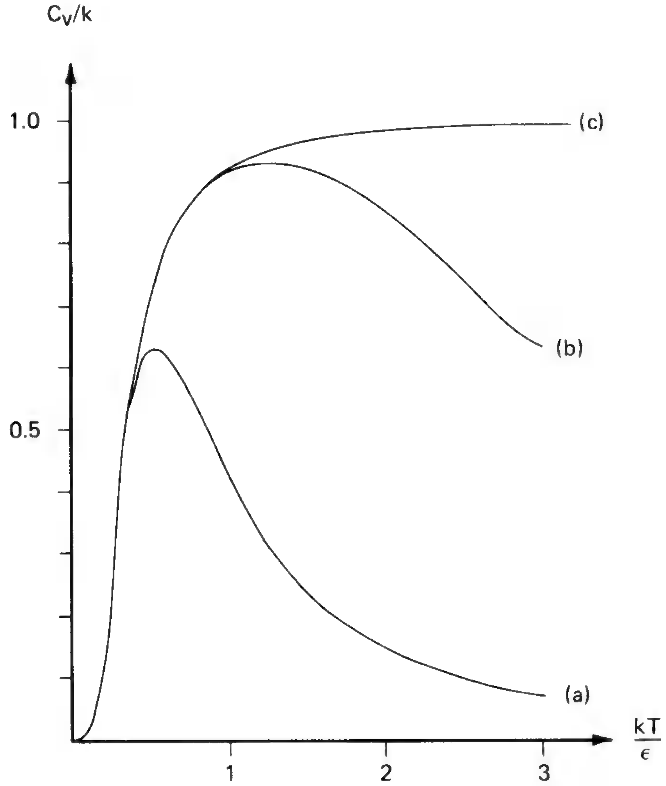
EXERCISE 25 – Randomly-Spaced Level System

The results are completely determined by which energy levels happen to be selected. The results below reflect a program selection of the following levels: 2.92092, 1.35865, 4.36366, 1.45409, and 1.99161.



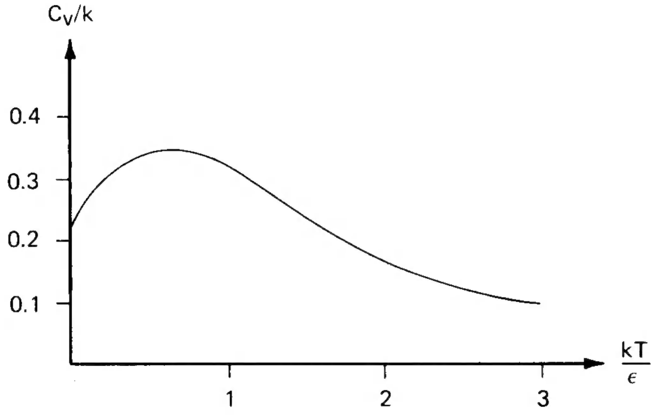
EXERCISE 27 – Heat Capacity, Equally-Spaced Level System

A graph of the results follows.



EXERCISE 29 – Heat Capacity, Randomly-Spaced Level System

The same energy levels were used as in Exercise 25. A graph of the results follows.



EXERCISE 31 – Two Systems, Different Temperature

System A has a higher average energy per oscillator than System B. When the two systems are brought into contact, quanta will be transferred back and forth between the two systems until the overall distribution is the Boltzmann distribution. The final temperature will be between T_A and T_B. Heat flow will take place from System A to System B.

CHAPTER FIVE

EXERCISE 33 – Entropy Calculation 1

```
1000  REM EXERCISE 33
1010  REM ENTROPY SIMULATION
1020  REM MAX SIZE IS 10 BY 10
1030  DIM B[10,10],C[10,10],F[30]
1035  DEF FNA(X)=(X+.5)*LOG(X)-X+.918939
1040  PRINT "INPUT SIDE OF SQUARE ARRAY";
1050  INPUT N1
1060  MAT C=CON[N1,N1]
1065  MAT B=(1)*C
1070  REM CARRY OUT TALLY
1080  GOSUB 2000
1100  REM RANDOMIZE RANDOM NUMBER GENERATOR
1110  PRINT
1120  PRINT "INPUT ANY INTEGER BETWEEN 1 AND 100";
1130  INPUT A
1140  FOR I=1 TO A
1150  LET X=RND(0)
1160  NEXT I
1170  PRINT
1175  PRINT
1180  PRINT "MOVES","ENTROPY/K"
1185  PRINT
1200  REM ENTER GAME LOOPS
1210  FOR M1=1 TO 11
1220  REM PRINT TALLY
1230  GOSUB 3000
1240  FOR N=1 TO 25
1250  REM MAKE  A MOVE
1260  GOSUB 4000
```

EXERCISE 33 – (Continued)

```
1270  NEXT N
1280  REM CARRY OUT TALLY
1290  GOSUB 2000
1300  NEXT M1
1310  REM END OF GAME LOOP
1320  STOP

2000  REM TELLY SUBROUTINE
2010  MAT F=ZER
2020  FOR I=1 TO N1
2030  FOR J=1 TO N1
2040  LET F[B[I,J]+1]=F[B[I,J]+1]+1
2050  NEXT J
2060  NEXT I
2070  REM END OF TALLY SUBROUTINE
2080  RETURN
3000  REM OUTPUT SUBROUTINE
3010  LET S=FNA(N1*N1)
3020  FOR I=1 TO 30
3030  IF F[I]=0 THEN 3050
3040  LET S=S-FNA(F[I])
3050  NEXT I
3060  PRINT (M1-1)*25,S
3070  REM END OF SUBROUTINE
3080  RETURN
4000  REM SUBROUTINE FOR MOVES
4010  LET I=INT(N1*RND(0)+1)
4020  LET J=INT(N1*RND(0)+1)
4030  LET K=INT(N1*RND(0)+1)
4040  LET L=INT(N1*RND(0)+1)
4050  IF B[I,J]=0 THEN 4010
4060  IF I#K THEN 4090
4070  IF J#L THEN 4090
4080  GOTO 4010
4090  LET B[I,J]=B[I,J]-1
4100  LET B[K,L]=B[K,L]+1
4110  REM END OF SUBROUTINE
4120  RETURN
9999  END
```

a) Equilibrium has been reached by about 150 moves.

```
RUN
MOVES             ENTROPY/K

 0                 0
 25                95.3125
 50                129.014
 75                137.848
 100               152.544
 125               155.916
 150               162.128
 175               165.19
 200               167.162
 225               170.536
 250               169.693
```

EXERCISE 33 – (Continued)

b) Equilibrium has been reached by about 75 moves.

```
RUN

MOVES          ENTROPY/K

 0              0
 25             94.5755
 50             113.294
 75             119.595
 100            123.244
 125            126.288
 150            126.299
 175            125.873
 200            127.685
 225            125.428
 250            124.322
```